VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2018

Bc. Luděk Páleník

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

OBECNÁ BEZPEČNOST INTERNETU VĚCÍ
GENERAL SECURITY OF THE INTERNET OF THINGS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Luděk Páleník**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Václav Oujezský, Ph.D.**
SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Luděk Páleník ***ID:*** 138004
Ročník: 2 ***Akademický rok:*** 2017/18

NÁZEV TÉMATU:

Obecná bezpečnost Internetu věcí

POKYNY PRO VYPRACOVÁNÍ:

V rámci diplomové práce využijte připravené zařízení Raspberry PI se zaměřením na sítě LoRa, LoRaWan. Sestavte, nainstalujte a uveďte do provozu Raspberry PI pro připojení k IoT (Internet of Things - Internetu věcí). V teoretické části popište sestavené zařízení, způsob zapojení pro IoT a LoRa a prozkoumejte bezpečnostní rizika této technologie. V praktické části se zaměřte na vybranou bezpečnostní slabinu a navrhněte a vytvořte vlastní řešení jejího testování za pomocí skriptu či algoritmu v jazyce Python.

DOPORUČENÁ LITERATURA:

[1] MONK, Simon. Programming the Raspberry Pi: getting started with Python. ISBN 978-0071807838

[2] “Building your own private LoRa network.” Building your own private LoRa network - Getting started with LoRa on mbed, [Online], Dostupné z: http://docs.mbed.com/docs/lora-with-mbed/en/latest/intro-to-lora/

Termín zadání: 5.2.2018 ***Termín odevzdání:*** 21.5.2018

Vedoucí práce: Ing. Václav Oujezský, Ph.D.
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Diplomová práce „Obecná bezpečnost internetu věcí" se v teoretické části zabývá rozborem sítí s nízkým odběrem elektrické energie – zejména sítěmi LoRaWAN. Dále se práce zabývá obecnou bezpečností Internetu věcí a také bezpečnostními riziky LoRaWAN sítí. Praktická část práce je zaměřena na sestavení brány LoRaWAN, která je realizována pomocí platformy Raspberry Pi s rozšiřujícím modulem iC880A-SPI. Brána je uvedena do provozu s kompletním nastavením, přiřazení síťovému serveru. Konfigurace čidel, vytvoření aplikace a jejich přiřazení aplikačnímu serveru. Dále je v praktické části realizován replay útok na technologii LoRaWAN, který demostruje zranitelnost této technologie.

KLÍČOVÁ SLOVA

Bash, Bezpečnost, IoT, iC880-SPI, Linux, LoRa, LoRaWAN, LPWAN, Python

ABSTRACT

The thesis "General Security of the Internet of Things" deals with the analysis of networks with low power consumption in the theoretical part – especially LoRaWAN networks. Further, the thesis deals with the general security of the Internet of things and the security risks of LoRaWAN networks. The practical part is focused on building the LoRaWAN gateway, which is realized with the Raspberry Pi platform with the iC880A-SPI extension module. The gateway is put into operation with full settings, assignment to the network server. Sensors configuration, creation an application, and assign it to the application server. In the practical part, a replay attack on LoRaWAN technology, which demonstrates the vulnerability of this technology, is realized.

KEYWORDS

Bash, IoT, iC880-SPI, Linux, LoRa, LoRaWAN, LPWAN, Security, Python

PÁLENÍK, Luděk. *Obecná bezpečnost Internetu věcí*. Brno, 2018, 92 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Václav Oujezský, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Obecná bezpečnost Internetu věcí“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Václavovi Oujezskému Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále bych také rád poděkoval společnosti Českomoravská světelná s.r.o., která zapůjčila LoRa čidla pro účely této práce. Velké poděkování patří také mé rodině, která mě podporovala po celou dobu studia.

Brno ....

podpis autora(-ky)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno ....

podpis autora(-ky)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM VÝPISŮ

ÚVOD

Tato diplomová se věnuje problematice nízkoenergetických sítí s dlouhým dosahem. Celý svět již dnes využívá různorodé technologie bezdrátového přenosu pro chytrá měření, vzdálenou správu, kontrolu pohybu atd. V počátku se tedy práce věnuje bezdrátovým technologiích s nízkým odběrem elektrické energie, kde jsou veškerá chytrá měřidla připojena k Internetu a podporuje tak myšlenku Internetu věcí a strojní komunikace. Jsou zde popsány nejznámější technologie, z nichž jedna je teprve na vzestupu a sice technologie LoRaWAN. Na konci této kapitoly je tabulkové srovnání popsaných technologií.

Cílem této práce je využití právě technologie LoRaWAN. Proto je druhá kapitola věnována modulaci LoRa, na které je technologie založena. Za účelem realizace kompletní LoRaWAN sítě je důležité znát veškeré aspekty dané technologie, proto je zde popsána také i její fyzická vrstva – LoRa modulace. Kapitola čtenáře seznamuje s inovativními faktory, které jsou v této technologii využity za účelem nezahlcování sítě, jako je například adaptivní datová rychlost jednotlivých koncových zařízení.

V třetí kapitole je LoRaWAN detailněji vyobrazena z hlediska síťových technologií. Popisuje topologii, architekturu, klíčové prvky, které jsou pro stavbu této sítě nezbytné. Dále seznamuje s rozdělením koncových zařízení do jednotlivých tříd, metody aktivace zařízení, kapacitu sítě a jak ji rozšířit. Na konci této kapitoly je popsáno, jakým způsobem je technologie LoRaWAN zabezpečena.

Čtvrtá kapitola se podrobně věnuje problematice obecné bezpečnosti Internetu věcí. Popisuje bezpečnostní cíle, které jsou strukturálně rozděleny. Jsou zde detailně popsány různé druhy útoků (fyzické, síťové, softwarové a šifrovací) na systém Internetu věcí. Je zde také seznam nejčastějších bezpečnostních problémů.

Kapitola pátá slouží jako obecné orientační schéma pro kapitolu následující. Je zde popsána analýza bezpečnostních rizik v sítích LoRaWAN. Jsou zde uvedeny důležité informace a popis jednotlivých aktiv v síti.

Kapitola šestá se věnuje již detailnímu popisu jednotlivých útoků, které jsou známy jako proveditelné v sítích LoRaWAN ve specifikaci 1.0.2.

Podrobnému popisu modulu iC880A-SPI, který je rozšiřujícím modulem pro platformu Raspberry Pi se věnuje sedmá kapitola. Jsou zde uvedeny úvodní informace o modulu, dále jeho vlastnosti a popis jednotlivých bloků. Je uvedeno, jaký způsobem lze modul připojit k Raspberry Pi. Je zde také detailní seznámení s procesorem SX1301, který je výpočetním prvkem celého modulu a obsahuje několik kanálu pro technologii LoRa.

Kapitola osmá popisuje instalaci systému Raspbian Stretch, který vychází z velmi dobře známé linuxové distribuce Debian. Po instalaci je uvedeno fyzické propojení modulu iC800A-SPI s platformou Raspberry Pi. Jsou zde přiloženy fotografie, které

znázorňují fyzické provedení celé brány. Následně je detailně popsána konfigurace zařízení Raspberry Pi jako je vytvoření nového uživatele, nastavení síťového rozhraní, vytvoření repozitáře atd. Dále je brána připojena k síťovému serveru, jsou nakonfigurována čidla, vytvořena aplikace na aplikačním serveru a realizována kompletní komunikace všech zařízení.

V poslední deváté kapitole je demonstrováno provedení Replay útoku, který je pro bližší seznámení detailně popsán v kapitole šesté. Útok je podrobně popsán a je proveden za pomocí zachycení a uložení komunikace LoRaWAN. Následně je napsán skript v jazyce Python, ve kterém jsou použita a replikována dříve zachycená data.

1 SÍTĚ S NÍZKÝM ODBĚREM ELEKTRICKÉ ENERGIE

IoT (Internet of Things) neboli česky „Internet věcí“ je síť objektů v masovém a globálním měřítku, která umožňuje připojit tyto objekty k síti tak, aby bylo možné sdílet informace prostřednictvím Internetu. Pro tento účel existuje mnoho technologií/norem pro připojení/přístup. Je využíváno kabelových, bezdrátových i satelitních technologií. Způsob přístupu závisí na povaze aplikace. Přístupové technologie s kabelovými např. optickými vlákny mají mimořádné výhody v oblasti bezpečnosti, spolehlivosti a nízké latence (v řádu nanosekund až mikrosekund), a proto jsou nejvhodnější pro aplikace, které vyžadují přísnou bezpečnost, spolehlivost a nízkou latenci. Bezdrátové technologie mají na druhou stranu zjevné výhody v mobilitě a zejména v ceně. Na rozdíl od satelitního systému bezdrátový systém vyvrací potřebu velkých antén a výkonných zesilovačů. Tyto výhody vedly k rostoucímu počtu průmyslových skupin, které pracují na vývoji nových technologií bezdrátových přístupových sítí určených pro aplikace IoT, u nichž je položení měděného drátu nebo optických vláken nepraktické a spolehlivost a latence jsou méně kritické [1].

LPWAN (Low Power Wide Area Network) jsou rozsáhlé sítě s nízkým odběrem elektrické energie, které jsou zacílené na IoT. Nabízí připojení k různým čidlům a akčním členům. Na rozdíl od tradičních mobilních širokopásmových sítí se tyto sítě nezaměřují na vysoké přenosové rychlostí a nízké latence, ale na škálovatelnost, rozšířené pokrytí, nízké náklady a energetickou účinnost koncových zařízení. Podle společnosti Cisco existuje již přibližně 20 miliard připojených zařízení a odhad na rok 2020 představuje více než 50 miliard připojených zařízení [2].

Ne všechna zařízení jsou připojena k síti LPWAN, ale k místním sítím, jako je Wi-Fi a BLE (Bluetooth Low Energy). Potenciál sítí LPWAN je však velmi významný. V dnešní době je ve většině částí světa hlavní platformou pro připojení k IoT existující GSM (Groupe Spécial Mobile) nebo GPRS (General Packet Radio Service) síť. Alternativním řešením pro GSM/GPRS je úzkopásmová NB-IoT (Narrow Band-Internet of Things) síť, která založená na standardu LTE (Long Term Evolution). NB-IoT bylo specifikováno s cílem nabídnout připojení IoT v pásmu 200 kHz v systému LTE. Kromě mobilních sítí jsou sítě IoT nasazovány i do průmyslových, vědeckých a lékařských pásem bez licence ISM (Industrial, Scientific and Medical). LPWAN zavádí myšlenku M2M (Machine to Machine) komunikace, tedy komunikace typu stroj-stroj. Tento druh komunikace byl vytvořen pro přenos informací v síti, a to bez lidské účasti [2, 3].

1.1 LTE-M

Společnost 3GPP (The 3rd Generation Partnership Project) zahájila studii o celulární komunikaci MTC (Machine Type Communications), která zkoumala požadavky na síť LTE. Ve verzi 12 byla zahrnuta omezení pro zařízení MTC (rozšířená MTC, často známá jako eMTC nebo formálně LTE-M). Tato omezení snižují náklady a spotřebu energie. Je aplikován zjednodušený hardware a úzkopásmový provoz (tj. 1,08 MHz), zatímco původní návrh LTE je zachován včetně přenosové struktury. Důvodem, proč LTE-M volí šířku pásma 1,08 MHz (šířka pásma šesti LTE zdrojových bloků) je, že existující kanál pro získávání LTE a kanál s náhodným přístupem mají šířku pásma 1,08 MHz (pevné a nekonfigurovatelné). Minimální přenosová šířka pásma UE (User Equipment) zůstává 180 kHz tj. jeden zdrojový blok, což je minimální plánovací jednotka staršího LTE pro zpětnou kompatibilitu. Rozšíření pokrytí o 15 dB, které odpovídá maximální ztrátě spojky 155 dB, je přidáno k LTE-M pro rozšíření pokrytí na oblasti, kde jsou nasazena zařízení MTC. Toto rozšíření umožňuje síti LTE dosáhnout na zařízení v místech s nadměrnými ztrátami pokrytí a je dosaženo časovými opakováními za cenu snížených přenosových rychlostí [1].

Předpokládalo se, že LTE-M by měl být vybudován v rámci stávajícího LTE a pro nízkoenergetickou IoT síť by byla nasazena 5G technologie. LTE-M proto zachovává minimální šířku pásma systému LTE 1,08 MHz a minimální šířku pásma komunikace 180 kHz na zařízení. Tato přenosová struktura má dva problémy:

- Pro rozšířené pokrytí je šířka pásma 180 kHz poměrně zbytečná, pokud jde o spektrální účinnost, což představuje vážný problém v masivních aplikacích IoT.
- Šířka pásma systému LTE-M o délce 1,08 MHz zabraňuje jeho zavedení do užších celulárních pásem, jako jsou pásma GSM.

GSM spektrum je rozděleno do kmitočtových pásem 200 kHz a v poslední době se stále více GSM spektra využívá pro nové bezdrátové služby. Tyto dva faktory znesnadňují, ne-li znemožňují podporu masivního nasazení IoT s ohledem na spektrum LTE. Technologie LTE-M je v licencovaném pásmu [1].

1.2 Narrow Band IoT

V září 2015 společnost 3GPP zahájila standardizační práci nazvanou NB-IoT (Narrow Band IoT), aby vyvinula nové vzdušné rozhraní speciálně upravené pro nízkoenergetická IoT zařízení. Cílem bylo zaměření především na využívání obnoveného GSM spektra. NB-IoT pracuje na šířce pásma 180 kHz, což se rovná starší šířce pásma zdroje LTE jak pro sestupné, tak pro vzestupné spojení. Zdrojem spektra může být obnovené GSM spektrum (200 kHz na nosič), zdrojový blok v pásmu LTE

nebo ochranné pásmo LTE nosiče. Sestupné spojení zachovává přenosovou strukturu LTE sestupného spojení – OFDMA (Orthogonal Frequency-Division Multiple Access) se vzdálenostní mezi nosnými částmi 15 kHz, zatímco vzestupné spojení je SC-FDMA (Single-carrier Frequency-Division Multiple Access) se vzdáleností mezi nosnými částmi 3,75 kHz (15 kHz pro vstupní pásmo, aby se zabránilo interferenci se sousedním starším LTE provozem). Technologie NB-IoT je v licencovaném pásmu [1, 3].

1.3 SIGFOX Ultra-Narrow Band

Technologie velmi úzkého pásma UNB (Ultra-Narrow Band) byla vyvinuta společností SIGFOX v roce 2009. Je to plošná technologie bezdrátového přístupu a poskytuje globální konektivitu pro nízkoenergetická IoT zařízení. UNB lze charakterizovat třemi klíčovými technologiemi:

- Úzkopásmové komunikační kanály (100 Hz, 600 Hz), které podporují ztrátu spojení více než 160 dB.
- Vzestupné přenosy, které se řídí modelem klient-server.
- Kooperativní příjem, tj. signál ze zařízení je společně přijímán několika základovými stanicemi UNB, aby se zvýšila pravděpodobnost úspěšného přijetí.

Kvůli přenosové struktuře klient-server však neexistuje centralizované plánování. Výhodou struktury klient-server je úspora pravidelného buzení pro provádění náročného postupu stránkování, čímž se maximalizuje životnost baterie. Takto se zařízení UNB probudí až v okamžiku, kdy dorazí data aplikací na vzestupné spojení. Poté se náhodně vybere kanál a data jsou odeslána. Pro zjednodušení přenosu je použita pevná velikost paketu 96 bitů na pevnou šířku pásma (např. 100 Hz). Přenos se třikrát opakuje, aby se zvýšila pravděpodobnost úspěšného příjmu. Po vzestupném přenosu následuje okno pro sestupný přenos dat na předem určeném kanálu. Technologie UNB je v bezlicenčním pásmu ISM [1, 2, 4].

1.4 Long Range

LoRa vznikla složením slov „Long Range“, v češtině „dlouhý dosah“ a je to proprietární technologie společnosti Semtech. Stejně jako SIGFOX je síť LoRa uzpůsobena do hvězdicové topologie, ve které je koncové IoT zařízení připojeno k centrálnímu síťovému serveru prostřednictvím koncentrátorů (bran). Minimální přenosová šířka pásma je 125 kHz, což podporuje maximální ztrátu spojky 157 dB. Proprietární energeticky účinná modulace LoRa založená na rozprostřeném spektru se používá ke zvýšení odolnosti proti rušení. Flexibilní přiřazení kanálů a adaptivní datové rychlosti

jsou podporovány za účelem optimalizace využití zdrojů a maximalizace kapacity. LoRa je, stejně jako SIGFOX UNB, v bezlicenčním pásmu ISM. V tabulce 1.1 lze shlédnout přehled jednotlivých popsaných technologií [1, 3, 4].

Technologie	**SIGFOX UNB**	**LoRa**	**LTE-M**	**NB-IoT**
Citlivost přijímače	–147 dBm	–137 dBm	–132 dBm	–137 dBm
Kmitočové pásmo	Sub-GHz ISM	Sub-GHz ISM	Licencované	Licencované
Min. přenos. šířka pásma	100 Hz, 600 Hz	125 kHz	180 kHz	3.75 kHz
Plně obousměrná	Ne	Ano	Ano	Ano
Modulace	D-BPSK[1]	LoRa, GFSK[2]	BPSK[3], QPSK[4], 16QAM[5], 64QAM	$\pi/2$ -BPSK, $\pi/4$ -QPSK
Přístup k médiu (MAC)	ALOHA[6]	ALOHA	SC-FDMA	SC-FDMA
Rychlost přenosu dat	100 b/s	0.3-38.4 kb/s	až 1000 kb/s	až 100 kb/s
Aktualizace vzduchem	Ne	Ano	Ano	Ano
Roaming	Ano	Ano	Ano	Ano
Standard	Ne	LoRaWAN	LTE (ver.12)	LTE (ver. 13)

Tab. 1.1: Srovnání LPWAN technologií [1, 3]

Cílem práce je zachytávání paketů v síti LoRa a je jí věnována v následujících kapitolách detailní pozornost.

[1] Differential Phase-Shift Keying – Diferenciální binární fázové klíčování.
[2] Gaussian Frequency-Shift Keying – Gaussovské frekvenční klíčování.
[3] Binary Phase-Shift Keying – Binární fázové klíčování.
[4] Quadrature Phase-Shift Keying –Kvadraturní fázové klíčování.
[5] Quadrature Amplitude Modulation – Kvadraturní amplitudová modulace.
[6] ALOHA je protokol s náhodným přístupem, který je v poslední době využíván zejména v MTC komunikaci [5].

2 LORA MODULACE

LoRa je fyzická vrstva nebo bezdrátová modulace používaná k vytvoření komunikačního spojení s dlouhým dosahem. Tato kapitola vychází ze zdrojů [6, 7, 8, 9, 10].

Mnoho starších bezdrátových systémů používá jako fyzickou vrstvu modulaci FSK (Frequency Shift Keying), protože je velmi účinná pro dosažení nízkého výkonu. LoRa® modulace je založena na CSS (Chirp Spread Spectrum) modulaci, tedy modulaci rozprostřeného spektra. Ta udržuje stejně nízké výkonové charakteristiky jako FSK, ale významně zvyšuje komunikační dosah.

CSS bylo používáno ve vojenské a vesmírné komunikaci po celá desetiletí díky dlouhým komunikačním vzdálenostem, které lze dosáhnout, a také kvůli robustnosti vůči rušení. LoRa je první nízkonákladová implementace pro komerční využití. Díky modulaci CSS může bezdrátové spojení dosáhnout citlivosti až -137 dBm a až 157 dB celkového rozsahu spojení. CSS používá různé faktory rozptýlení SF (Spreading Factor) od SF7 do SF12, což nejen zvyšuje kapacitu sítě, ale také umožňuje dynamické přizpůsobení datových rychlostí zařízení. Zařízení s lepším propojením na bránu (díky blízkosti nebo prostředí s nižším šumem) mohou využívat vyšší přenosové rychlosti (až 11 kb/s) a šetřit baterii.

Zařízení se špatnou kvalitou spojení mohou zvýšit celkový rozsah spojení pomocí nižších datových rychlostí a rozšířit tak efektivní připojení na více než 30 km v přímém pohledu. Pokud je hodnota SF zvýšena, velikost přenášených dat bude snížena, což bude mít za následek větší výkon na kanál a delší komunikační vzdálenost. Tabulka 2.1 uvádí rozptylové faktory LoRa pro šířku pásma 125 kHz. Jak lze pozorovat, je-li faktor rozptýlení zvýšený, sníží se rychlost datového toku a prodlužuje se časový limit, ale značně se zlepšuje limit SNR (Signal-to-noise ratio) neboli odstup signál-šum. Počet kódovaných symbolů se snižuje se zvyšujícím se faktorem rozptýlení.

Faktor rozptýlení	Počet symbolů [s]	SNR [dB]	Čas na vzduchu pro 10 B paket [s]	Datová rychlost [bit/s]
7	976	-7,5	56	5469
8	488	-10	103	3125
9	244	-12,5	205	1758
10	122	-15	371	977
11	61	-17,5	741	537
12	30	-20	1483	293

Tab. 2.1: LoRa faktory rozptýlení pro pásmo 125 kHz

LoRa® modulaci lze použít v širokém rozsahu frekvencí od 137 MHz do 1020 MHz. To zahrnuje řadu ISM bezlicenčních pásem, jako jsou 169 MHz, 433 MHz, 868 MHz a 915 MHz. Toto je klíčový nástroj pro levné, celosvětové nasazení a interoperabilitu.

Hlavní výhodou je integrační schopnost velkého počtu uzlů. Pokud je uzel umístěn blíže k bráně, bude mít vyšší komunikační přenosová rychlost díky integrovanému mechanismu ADR (Adaptive Data Rate). Tento mechanismus umožňuje projektantovi optimalizovat výkon sítě v konstantní šířce pásma.

LoRa je implementace fyzické vrstvy a je agnostická s implementacemi na vyšší vrstvě. To umožňuje, aby LoRa koexistovala a spolupracovala s existujícími síťovými architekturami. Kompromisem je dosažitelná datová rychlost, která je v rozsahu kilobitů za sekundu. Ačkoli není LoRa® vhodná pro streamování videa, je vhodná pro službu IoT a aplikace M2M.

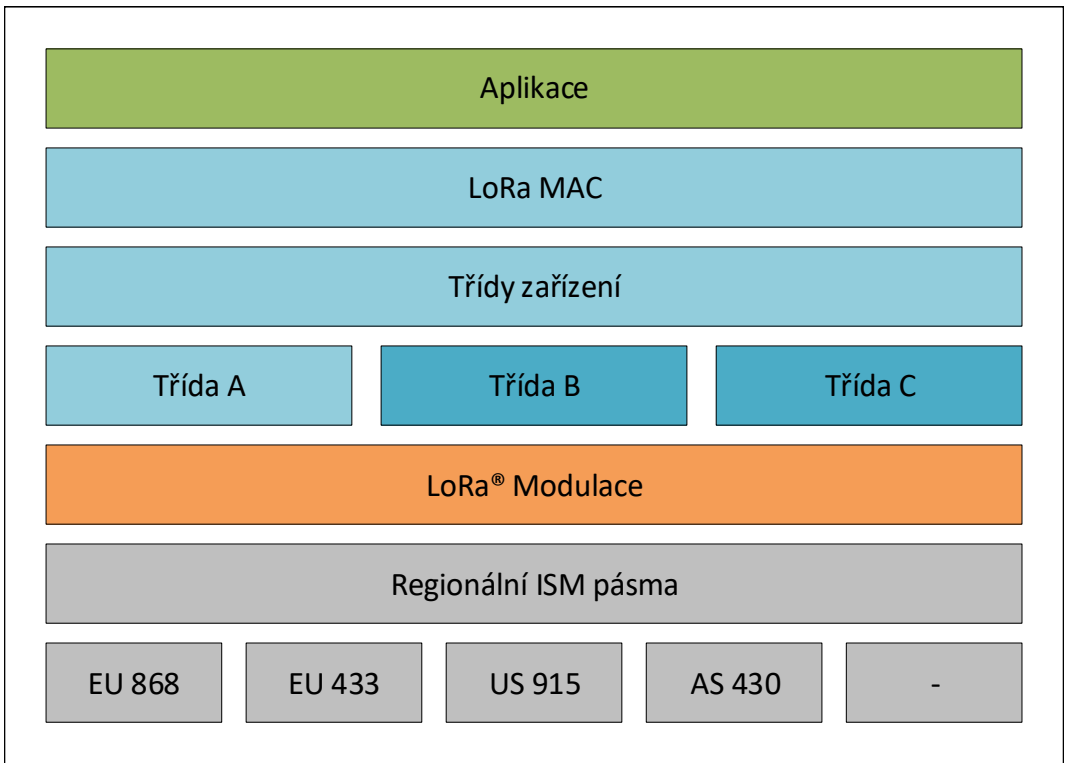
2.1 Klíčové vlastnosti LoRa modulace

1. **Škálovatelnost šířky pásma**
   V LoRa modulaci lze škálovat kmitočet i šířku pásma. Může být použita jak pro úzkopásmové frekvenční skoky, tak pro širokopásmové přímé sekvenční aplikace. Na rozdíl od stávajících úzkopásmových nebo širokopásmových modulačních schémat lze LoRa modulaci snadno přizpůsobit pro jakýkoli provozní režim, pouze s několika konfiguračními změnami.
2. **Konstatní obálka/nízký odběr energie**
   Podobně jako u FSK má LoRa modulace konstantní schéma obálky. Nicméně LoRa může snížit výstupní výkon vysílače ve srovnání s běžným FSK při zachování stejného nebo lepšího rozsahu spojení, a tím navíc šetřit odběr elektrické energie daného zařízení.
3. **Odolnost vůči vadnutí spektra**
   Pulz Chirp modulace je poměrně širokopásmový, takže LoRa nabízí odolnost vůči vadnutí spektra.
4. **Vysoká interferenční odolnost**
   Kvůli vysokému koeficientu BT (Bandwith Time product) a jeho asynchronnímu charakteru je signál LoRa velmi odolný vůči interferenčním mechanismům v pásmu i mimo pásmo.
5. **Rezistence vůči Dopplerově jevu**
   Dopplerův jev způsobuje malý posun frekvence v impulsu LoRa, který zavádí relativně zanedbatelný posun v časové ose signálu základního pásma. Tato tolerance frekvenčního offsetu snižuje požadavek na zdroje přesných referenčních hodin.

6. **Velký rozsah**
   Pro pevný výstupní výkon a propustnost je rozsah spojení LoRa vyšší než konvenční FSK.
7. **Zvýšená síťová kapacita**
   Semtech LoRa modulace využívá ortogonální rozptylovací faktory, které umožňují vysílat více rozprostřených signálů ve stejnou dobu a na stejném kanálu bez minimální degradace citlivosti.
8. **Rozsah/lokalizace**
   Vlastností LoRa je schopnost lineárně rozlišovat mezi frekvenčními a časovými chybami. LoRa je ideální modulace pro radarové aplikace a je tedy vhodná pro lokalizační aplikace, jako jsou služby určování polohy v reálném čase.

3 LORAWAN

LoRaWAN (Long Range Wide Area Network) definuje komunikační protokol a architekturu systému pro síť, zatímco fyzická vrstva LoRa umožňuje komunikaci na dlouhou vzdálenost. Protokol a síťová architektura mají největší vliv na určení životnosti baterie uzlu, kapacity sítě, kvality služeb, zabezpečení a rozmanitosti aplikací obsluhovaných sítí. Rozložení vrstev LoRaWAN lze vidět na obrázku 3.1. LoRaWAN je určena pro bezdrátová zařízení, napájena pomocí baterií a síť může být v regionálním, národním nebo globálním měřítku. Zaměřuje se na klíčové požadavky týkající se Internetu věcí, jako jsou například bezpečná obousměrná komunikace, mobilita a lokalizační služby. Specifikace LoRaWAN poskytuje bezproblémovou interoperabilitu mezi inteligentními věcmi bez nutnosti komplexních lokálních instalací a dává svobodu uživateli, vývojářům a podnikům umožnit zavádění Internetu věcí. Informace v této kapitole jsou čerpány ze zdrojů [6, 7, 8, 9], pokud není uvedeno jinak.

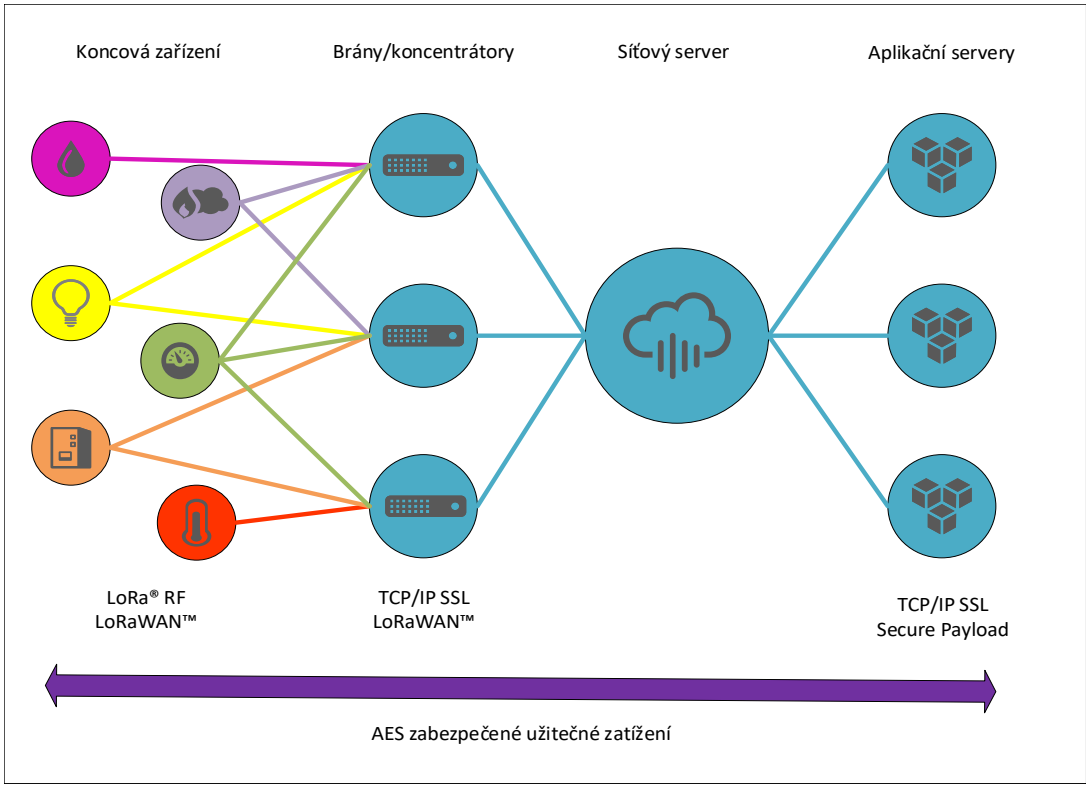


Obr. 3.1: Vrstvový model LoRaWAN [6]

3.1 Síťová architektura

Mnoho stávajících nasazených sítí využívá smíšenou topologii sítě, známou jako „Mesh“. V této topologii předávají jednotlivé koncové uzly informace o dalších uzlech, čímž zvyšují rozsah komunikace a velikost buněk v síti. Snižuje se tak síťová kapacita a životnost baterie, protože uzly přijímají a předávají informace z jiných uzlů, které jsou pro ně pravděpodobně irelevantní. LoRaWAN síťová architektura je typicky rozmístěna v hvězdicové topologii. Dlouhý dosah hvězdicové architektury má největší smysl pro zachování životnosti baterie a připojení na velkou vzdálenost.

V síti LoRaWAN nejsou uzly přidruženy ke konkrétní bráně. Namísto toho jsou data, přenášená uzlem, typicky přijímána více branami. Brány zde fungují jako transparentní most, který předává zprávy mezi koncovými zařízeními a centrálním síťovým serverem. Brány jsou připojeny k síťovému serveru prostřednictvím standardních IP (Internet Protocol) připojení, zatímco koncová zařízení používají bezdrátovou komunikaci s LoRa rádiovým připojením. Veškerá komunikace koncových bodů je obecně obousměrná, ale také podporuje operace, jako je vícesměrové vysílání. Obecná architektura LoRaWAN je znázorněna na obrázku 3.2.



Obr. 3.2: Architektura LoRa sítě [6]

3.2 Komponenty LoRaWAN sítě

Síť LoRaWAN se skládá z několika prvků [11]:

- **Koncová zařízení** představují nejvíce periferních prvků sítě. Mohou být vybaveny různými typy digitálních senzorů a firmwarem, který je napsán v jazyce C jako monolitický kus kódu. Kód je kompilován a přemístěn do paměti přístroje. Pakety generované uzly jsou vysílány a mohou být zaslechnuty libovolnou branou v rozsahu pokrytí.
- **Brány** jsou zařízení mezilehlých uzlů, které přijímají přenosy z uzlů pomocí LoRa modulace. Do paketu přidávají další informace (např. týkající se síly přijatého signálu) a předávají přijaté pakety na síťový server pomocí IP. Užitečná data, která brána odesílá na síťový server, má strukturu JSON (JavaScript Object Notation).
- **Síťový server** je v každé LoRaWAN síti pouze jeden. Implementuje zásobník protokolů LoRa až po vrstvu MAC (Medium Acces Control), aby hladce komunikoval s uzly (přes brány). Síťový server navíc podporuje další komunikační protokoly. Představuje místo přístupu do sítě zvenčí. Server je zodpovědný za nastavení parametrů MAC, které mají být použity v uzlech LoRa v přidružené síti. Server také podporuje základní rozhraní pro zprostředkovatele MQTT (Message Queuing Telemetry Transport), aby exportoval data shromážděná pomocí uzlů LoRa a přijímal příkazy pro koncové uzly. Síťový server tak poskytuje rozhraní, které umožní komunikaci s uzly LoRa a aplikačním serverem.
- **MQTT zprostředkovatel** je server, který shromažďuje příkazy MQTT publikování/odběr a odesílá data odpovídajícím způsobem. Poskytuje standardní autentizační mechanismy, které umožňují přístup k datům pouze oprávněným uživatelům.
- **Aplikační server** je server, který se přihlásí ke všem zdrojům, které publikuje určitý síťový server. Aplikační server může diskriminovat aplikaci a uzel LoRa, který generuje všechna data pomocí struktury adresování MQTT. Proto jsou všechna data, přijatá od MQTT zprostředkovatele, zpracovávána podle konkrétní aplikace. Server proto hostí více entit – jeden pro každý typ aplikace běžící na přidružených sítích. Tyto entity jsou společně navrženy s přidruženými aplikacemi a budou schopny správně interpretovat zprávy, generované každým uzlem LoRa, a poskytovat tak standardní rozhraní API (Application Programming Interface) pro interakci s různými aplikacemi. Rozhraní API bude založeno na MQTT s kódováním JSON.

3.3 Kapacita sítě

Aby síť založená na hvězdicové topologii s dlouhým dosahem byla životaschopná, tak musí mít brána velmi vysokou kapacitu neboli schopnost přijímat zprávy z velmi velkého objemu uzlů. Vysoká kapacita v síti LoRaWAN je dosažena použitím adaptivní přenosové rychlosti a multikanálového multimodálního vysílače v bráně, takže lze přijímat simultánní zprávy na více kanálech. Kritickým faktorem ovlivňujícím kapacitu je počet souběžných kanálů, rychlost přenosu dat, délka užitečných dat a kmitočet uzlů.

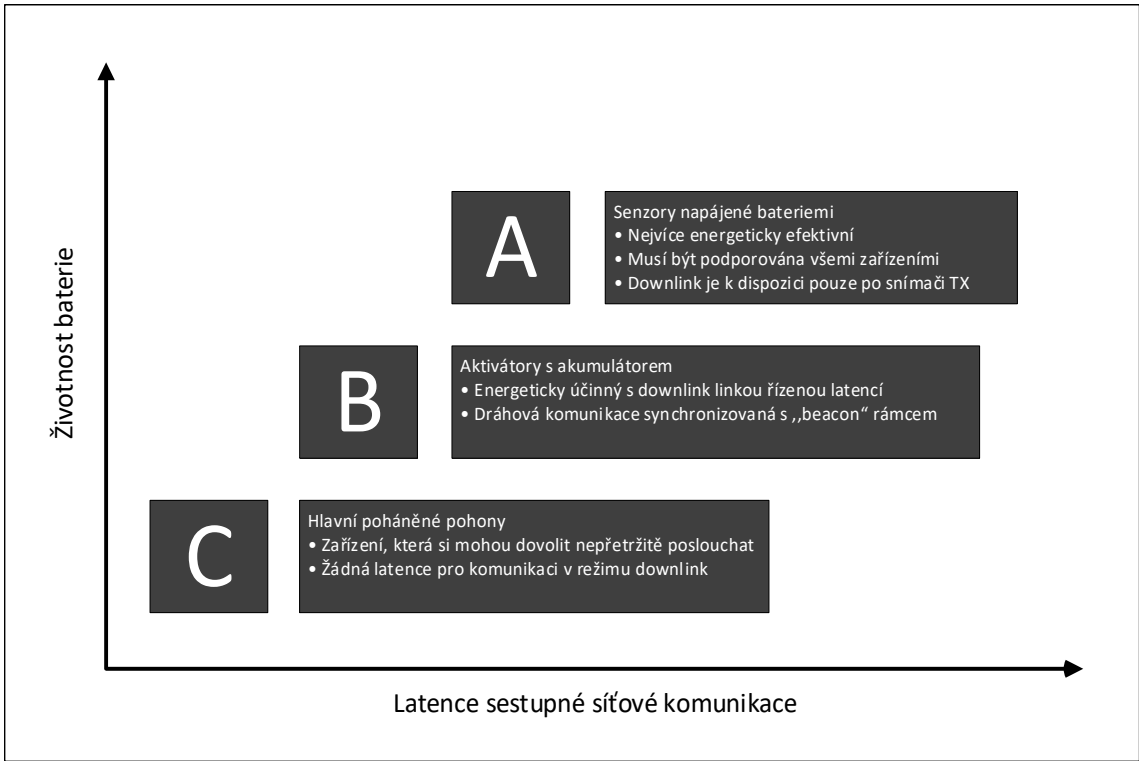
LoRa modulace používá různé faktory rozprostření, takže jsou signály na sebe prakticky kolmé. Při změně faktoru rozprostření se mění i účinná rychlost přenosu dat. Brána využívá tuto vlastnost tím, že může současně přijímat více datových rychlostí na stejném kanálu. Pokud má uzel kvalitní připojení a je blízko k bráně, není důvod, aby vždy používal nejnižší datovou rychlost. Změnou rychlosti přenosu dat se zkrátí čas na vzduchu, což otevírá další potenciální prostor pro přenos dalších uzlů. Adaptivní přenosová rychlost také optimalizuje životnost baterie uzlu. Za účelem práce s adaptivní datovou rychlostí je zapotřebí symetrické vzestupné i sestupné spojení s dostatečnou kapacitou pro sestupné spojení. Tyto funkce umožňují síti LoRaWAN dosáhnout velmi vysoké kapacity a škálovatelnosti. Síť může být nasazena s minimálním množstvím infrastruktury. Pro zvýšení kapacity lze přidat více bran, snížit tak množství přenosu na jiné brány a měnit celkovou kapacitu sítě.

3.4 Třídy zařízení

Koncová zařízení slouží různým aplikacím a mají různé požadavky. Pro optimalizaci různých profilů koncových aplikací využívá LoRaWAN různé třídy zařízení. Třídy zařízení kompenzují latenci komunikace ve směru sestupného spojení v závislosti na životnosti baterie. U aplikací řídicího typu je zpoždění ve směru sestupného spojení důležitým faktorem. Srovnání zařízení na obrázku 3.3.

- Obousměrná koncová zařízení (třída A):
  Koncová zařízení třídy A umožňují obousměrnou komunikaci, při níž každý vzestupný přenos každého koncového zařízení je následován dvěma krátkými přijímacími okny pro sestupné spojení. Přenosový slot naplánovaný koncovým zařízením je založen na vlastních komunikačních potřebách s malou variací založenou na náhodném čase. Třída A je systém pro koncová zařízení s nejnižší výkonností. To zahrnuje aplikace, které vyžadují pouze sestupnou komunikaci ze serveru v momentě, kdy koncové zařízení odeslalo přenos na vzestupné spojení. Sestupná komunikace ze serveru v jakémkoliv jiném čase bude muset počkat až na další naplánovaný vzestupný přenos.

- Obousměrná koncová zařízení s plánovanými přijímacími sloty (třída B): Koncová zařízení třídy B v plánovaných časech otevírají další přijímací okna. Aby koncové zařízení mohlo v plánovaném čase otevřít své přijímací okno, přijímá z brány časově synchronizovaný „beacon“ rámec. To umožňuje serveru zjistit, kdy poslouchá koncové zařízení.
- Obousměrná koncová zařízení s maximálními přijímacími okny (třída C): Koncová zařízení třídy C mají téměř nepřetržitě otevřená přijímací okna, která jsou zavřená pouze při vysílání.



Obr. 3.3: Srovnání třídy zařízení v závislosti na životnosti baterie a latenci sestupného spojení [6]

3.4.1 Aktivace zařízení

U LoRa zařízení je potřebná jejich aktivace, aby tato zařízení mohla být připojena do sítě. Existují dvě metody aktivace [12]:

1. **Over-the-Air-Activation**
   OTAA (Over-the-Air-Activation) aktivace je založena na základě globálně jedinečného identifikátoru. Aktivace probíhá pomocí definovaného „handshake“,

nebo-li potřesení rukou při navázání komunikace. Koncové zařízení přenáší žádost o připojení do aplikačního serveru (Join Request). Tato žádost obsahuje následující informace:

- globálně jedinečný identifikátor koncových zařízení (DevEUI),
- identifikátor aplikace (AppEUI),
- ověření pomocí aplikačního klíče (AppKey).

Koncové zařízení obdrží zprávu z aplikačního serveru (Join Accept). Tato zpráva je koncovým zařízením následně autentizována a dešifrována. Zařízení extrahuje a ukládá adresu zařízení (DevAddr), kterou vygeneroval aplikační server. V posledním kroku jsou uloženy klíče dané relace:

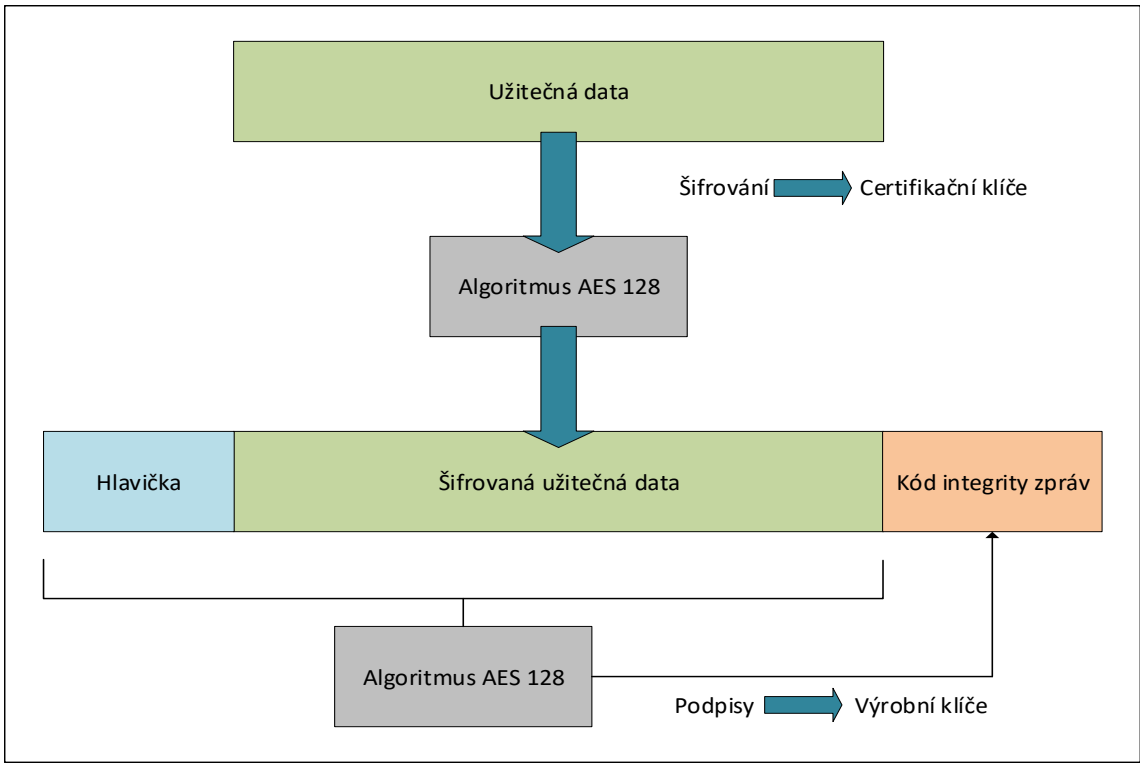
- klíč síťové relace (NwkSKey),
- klíč aplikační relace (AppSKey).

2. **Activation By Personalization**
   ABP (Activation By Personalization) aktivace je založena na uložení platných informací přímo do zařízení. Následující informace jsou nakonfigurovány již z výroby:
   - adresa zařízení (DevAddr),
   - klíč síťové relace (NwkSKey),
   - klíč aplikační relace (AppSKey).

   Při tomto druhu aktivace nedochází k žádnému „handshake“. Výrobní informace je možné změnit za informace, které nově vygeneruje server, ale je k tomu zapotřebí fyzický přístup ke koncovému zařízení.

3.5 Bezpečnost

Bezpečnost je pro budoucí sítě IoT velmi důležitá, protože zajistí provoz bez vnějšího narušení. LoRaWAN zvažuje dvě vrstvy zabezpečení, jednu pro síť a druhou pro aplikace, jak je znázorněno na obrázku 3.4. Řešení sítě LoRaWAN přichází s rámcem ověřovacím a bezpečnostním a s rámcem založeným na šifrovacím algoritmu AES-128 (Advanced Encryption Standard). AES-128 šifruje rámec pro zachování důvěryhodnosti a generuje MIC (Message Integrity Code) pro integritu. Každé koncové zařízení má přiřazený klíč od výrobce zařízení nebo od vlastníků aplikací. Ověřování a šifrování jsou odděleny, takže je možné autentizovat pakety a zajistit ochranu integrity [6, 7].



Obr. 3.4: Schéma zabezpečení algoritmem AES-128 [6, 7]

4 OBECNÁ BEZPEČNOST INTERNETU VĚCÍ

Tato kapitola popisuje obecnou bezpečnost Internetu věcí a její majoritní část vychází ze zdroje [13].

Internet věcí umožňuje elektronickým zařízením v okolním prostředí sdílet informace s dalšími členy (zařízeními) sítě, které umožňují rozpoznat události a změny v jejich okolí a jednat a reagovat autonomně bez jakékoli lidské interakce. Výhody IoT jsou rozsáhlé a jeho aplikace mění způsob, jak lidé pracují a žijí. Zejména tak, že šetří čas a zdroje a otevírá nové příležitosti pro růst, inovace a výměnu informací mezi entitami.

Existence rozsáhlé IoT sítě propojených zařízení však představuje nové ohrožení bezpečnosti, ochrany soukromí a důvěryhodnosti, která ohrožuje všechna tato zařízení vysokým rizikem, a tím poškozuje přidružené uživatele. Internet je základem a jádrem podporujícím IoT, takže téměř všechny bezpečnostní hrozby, které spadají do Internetu, se propagují i do Internetu věcí. Navíc rychlý vývoj a širší přijetí zařízení IoT v našich životech znamená naléhavost řešení těchto bezpečnostních hrozeb před nasazením.

Přestože řada společností tvrdí, že jejich technologie jsou zabezpečeny a chráněny, jsou stále náchylné k různým typům útoků. Vzhledem k tomu, že propojená zařízení mají přímý dopad na život uživatelů, existuje potřeba správně definované klasifikace bezpečnostních rizik a správné bezpečnostní infrastruktury s novými systémy a protokoly, které mohou zmírnit bezpečnostní problémy týkající se soukromí, integrity dat a dostupnosti v Internetu věcí.

Termín Internet věcí byl poprvé představen v roce 1999 Kevinem Ashtonem. Myšlenka Internetu věcí je dnes však reálným faktem, který propojuje reálné světové senzory, elektronická zařízení a systémy s Internetem.

- Spotřebitelské služby, inteligentní domy a inteligentní objekty.
- Chytré měření energií; inteligentní měřiče a sítě.
- Chytré telefony a tablety.
- Vozidla připojená k síti Internet.
- Zařízení pro monitorování zdraví, chytré hodinky, inteligentní oblečení, inteligentní obojky s implantovanými RFID (Radio Frequency Identification) identifikátory a dokonce i lidské implantované přístroje (kardiostimulátory).
- Bezdrátové senzorové sítě: měření počasí, průmyslové monitorování, protokolování dat, monitorování životního prostředí (kvalita vody, detekce požáru, monitorování znečištění ovzduší) atd.

4.1 Bezpečnostní cíle

Protože IoT je relativně nový koncept, je třeba definovat jeho bezpečnostní cíle. IoT je implementace síťových technologií a integrace stávajících síťových infrastruktur. Proto jsou všechny bezpečnostní problémy a hrozby každé síťové technologie předány standardně do systému IoT, který využívá tyto technologie. Dále existuje možnost dalších bezpečnostních hrozeb, které vyplývají z koexistence a spolupráce různých technologií a otevřených standardů a protokolů vytvořených pro internet věcí. Nejžádanějším bezpečnostním cílem internetu věcí je ochrana shromážděných dat, protože data shromážděná z fyzických zařízení mohou také obsahovat citlivé informace o uživateli. Z tohoto důvodu musí být bezpečnost jakéhokoli systému IoT odolná vůči útokům souvisejícím s daty.

4.1.1 Bezpečnost a soukromí v IoT

Bezpečností a soukromím se rozumí ochrana jakýchkoli shromážděných nebo uložených dat v jakémkoli systému IoT. To znamená, že systém IoT musí v každém okamžiku poskytovat důvěrnost údajů, integritu a dostupnost. Toho lze dosáhnout použitím ověřování, řízením přístupu, šifrováním a zajištěním dostupnosti dat, zálohováním dat atd.

4.1.2 Důvěra

Důvěra je komplikovaný koncept skládající se z různých vlastností a cílů. Důvěra se týká prosazováním výše uvedených bezpečnostních cílů, které spočívají v následujících bodech:

1. **Důvěryhodné vztahy mezi jednotlivými vrstvami IoT**
   Správná komunikace a přechod mezi různými vrstvami systému IoT a mezi různými existujícími uzly jsou potřebné pro zajištění bezpečnosti a soukromí dat.
2. **Důvěra v zabezpečení a soukromí v každé vrstvě IoT**
   Zabezpečení a ochrana osobních údajů v každé vrstvě systému IoT musí fungovat nepřetržitě. Jen tak je možné zajistit spolehlivost, integritu a důvěrnost údajů.
3. **Důvěra mezi uživatelem a systémem IoT**
   V určitém okamžiku může systém IoT zveřejnit některé údaje koncovému uživateli a naopak. Proto je potřeba poskytnout důvěru mezi systémem a koncovým uživatelem pro úspěšné uplatnění koncepce IoT. Kromě toho bude uživatel interagovat se systémem IoT a jeho akce mohou mít vliv na správnost fungování systému. Existuje tedy potřeba klasifikovat akce uživatele.

4.2 Klasifikace bezpečnostních útoků na IoT

Existuje široké spektrum bezpečnostních rizik a útoků v systémech Internetu věcí. Tato podkapitola rozděluje různé útoky do čtyř odlišných tříd – fyzické, síťové, softwarové a šifrovací útoky. IoT je implementováno pomocí různých stávajících síťových technologií. Existuje tedy potřeba správné kategorizace útoků tak, aby zapouzdřily všechny různé druhy hrozeb.

4.2.1 Fyzické útoky

Tyto útoky se zaměřují na hardwarové komponenty systému IoT. Útočník musí být fyzicky blízko uzlu nebo celého systému. Jsou zde zahrnuty i útoky, které poškozují životnost nebo funkčnost hardwaru.

1. **Manipulace s uzlem**
   Útočník může způsobit poškození senzorového uzlu tím, že fyzicky znehodnotí celý uzel nebo jeho část. Dále např. dotazování uzlů, za účelem získání citlivých informací (sdílené kryptografické klíče nebo směrovací tabulky).
2. **Rušení v kmitočtovém pásmu RFID**
   Útok na odepření služby může být implementován na libovolném identifikátoru RFID vytvořením a odesláním šumových signálů přes signály rádiové frekvence, které RFID používají pro komunikaci. Šumové signály mohou rušit signály RFID a bránit tak komunikaci.
3. **Uzamčení uzlu v bezdrátové senzorové síti**
   Ekvivalentní útok k RFID rušení v bezdrátových senzorových sítích. Útočník může zasahovat do rádiových frekvencí uzlů bezdrátového senzoru, zablokovat signály a přerušit komunikaci k uzlům. Pokud se útočníkovi podaří zablokovat uzly, může znehodnotit danou IoT službu.
4. **Nasazení škodlivého uzlu**
   Protivník může fyzicky nasadit nový škodlivý uzel mezi dvěma nebo více uzly systému IoT, a tím řídit veškerý tok dat z uzlů a do uzlů a sledovat jejich provoz. Tento typ útoku je znám jako „Man in the middle“ nebo-li „muž uprostřed“.
5. **Fyzické škody**
   Protivník může fyzicky poškodit zařízení sítě IoT pro svůj vlastní zisk. Tento druh útoku se liší od útoku „Manipulace s uzlem“, protože v této situaci se protivník snaží přímo poškodit systém IoT s cílem ovlivnit dostupnost služby.

6. **Sociální inženýrství**
   Útočník manipuluje uživatele systému IoT a získává tak např. citlivé informace. Tento druh útoku je zařazen do kategorie fyzických útoků, protože útočník musí fyzicky komunikovat s uživateli sítě IoT, aby dosáhl svých cílů.
7. **Nedostatek „spánku“**
   Velké množství senzorů v systému IoT je napájeno pomocí baterií. Senzory, v závislosti na technologii, odesílají data v určitých intervalech a zbytek času jsou v podstatě v určitém typu úsporného režimu. Tento typ útoku ale nutí uzly udržovat neustálé vzestupné nebo sestupné spojení, což vede k vyšší spotřebě energie a způsobí, že se uzly po určité době vypnou.
8. **Nasazení škodlivého kódu**
   Útočník kompromituje uzel tím, že do něj fyzicky implementuje škodlivým kód, který mu následně umožní např. přístup k systému IoT či záznam citlivých dat.

4.2.2 Síťové útoky

Tyto útoky jsou zaměřeny na systémovou síť a útočník nemusí být nutně blízko k systémové síti, aby útok mohl aplikovat.

1. **Analýza provozu**
   Útočník může zachytit důvěrné informace nebo jakékoli jiné údaje. Také téměř u všech útoků se útočník nejprve pokusí získat některé informace o síti dříve, než aplikuje svůj útok. To se provádí např. pomocí aplikací pro snímání portů, vytváření paketů atd.
2. **RFID Spoofing**
   Útočník může zachytit komunikaci RFID se čtečkou a zjistit údaje. Poté může poslat své vlastní údaje obsahující původní identifikátor štítku, takže se zdá, že je platný. Útočník získá plný přístup k systému, protože vůči systému se jeví jako původní nosič údajů.
3. **Zahození komunikace**
   Útočník přesměruje veškerou komunikaci ze senzorových sítí na virtuální záchytný bod. Tento typ útoku spočívá v zahození veškerých paketů, které jednotlivé senzory ze sítě odešlou na tento bod. Celý systém tak může začít kolabovat, protože neobdržuje data, která mohou rozhodovat o dalším automatizovaném řízení.
4. **Útok „Muž uprostřed“**
   Útočník v síti dokáže zasahovat mezi dvěma uzly, přistupovat k omezeným datům, porušovat soukromí obou uzlů monitorováním, odposloucháním a ovládáním komunikace. Na rozdíl od kategorie „Nasazení škodlivého kódu“ z katego-

rie fyzických útoků nemusí útočník fyzicky existovat, aby byl tento typ útoku úspěšný, ale spoléhá výhradně na síťové komunikační protokoly systému IoT.

5. **Odmítnutí služby**
   Útočník může zatěžovat síť IoT s většími provozními daty, což může vést k úspěšnému útoku odmítnutí služby z důvodu zahlcení síťového rozhraní.
6. **Útok na směrovací informace**
   Jedná se o přímé útoky na síť, které mohou např. měnit záznamy ve směrovacích tabulkách, vytvořit smyčky směrování, povolit nebo zakázat provoz, odesílat falešné chybové zprávy.
7. **Útok závadného uzlu**
   Závadný uzel je jediný uzel, který nárokuje identitu většího počtu uzlů, za které se vydává. Tento druh útoku vede k tomu, že sousední uzly senzorové sítě akceptují nepravdivé informace.

4.2.3 Softwarové útoky

Softwarové útoky jsou hlavním zdrojem bezpečnostních rizik v jakémkoli počítačovém systému. Softwarové útoky zneužívají systém pomocí programů trojských koní, červů, virů a škodlivých skriptů, které mohou ukrást informace, manipulovat s daty, popřít služby a dokonce poškodit zařízení systému IoT.

1. **Phishingové útoky**
   Útočník získá přístup k důvěrným údajům tím, že znehodnotí autentizační proces ověření uživatele, obvykle prostřednictvím infikovaných e-mailů nebo webových stránek.
2. **Viry, červi, trojští koně**
   Útočník může infikovat systém škodlivým softwarem, což má za následek např. krádež informací, narušení nebo dokonce odmítnutí služby.
3. **Škodlivé skripty**
   Obvykle je síť IoT připojena k Internetu. Uživatel, který řídí bránu, může být nasměrován ke spustitelných skriptům, což může vést k úplnému vypnutí systému nebo ke krádeži dat.
4. **Odmítnutí služby**
   Útočník může provést útoky DoS (Denial of Service) nebo distribuované odmítnutí služby DDoS (Distributed Denial of Service) na síť IoT prostřednictvím aplikační vrstvy, která postihuje všechny uživatele v síti. Tento druh útoku může také zablokovat oprávněné uživatele z aplikační vrstvy a poskytnout plné řízení aplikační vrstvy útočníkovi, který bude mít přístup k databázi a citlivým soukromým údajům.

4.2.4 Šifrovací útoky

Tyto útoky jsou založeny výlučně na prolomení šifrovacího schématu používaného v systému IoT.

1. **Útoky postranním kanálem**
   Pomocí specifických technik může útočník na šifrovacích zařízeních systému IoT načíst šifrovací klíč používaný pro šifrování a dešifrování dat.
2. **Kryptoanalýza**
   Tyto útoky předpokládají držení šifrovaného textu/dat a jejich účelem je najít šifrovací klíč, který se používá. Poté je možné prolomení šifrovaných dat v systému.
3. **Útok „Muž uprostřed“** Pokud si dva uživatelé systému IoT **A** a **B** vymění klíče během vytváření zabezpečeného komunikačního kanálu, protivník se mezi nimi umístí na komunikační lince. Protivník pak může zachycovat signály, které **A** a **B** vysílají. Pokusí se zasahovat tak, že provede výměnu klíčů s **A** a **B** odděleně . Protivník pak bude schopen dešifrovat/šifrovat všechna data pocházející z **A** a **B** pomocí klíčů, které sdílí s oběma. **A** a **B** si budou myslet, že mezi sebou komunikují.

4.3 Nejčastější zranitelnosti Internetu věcí

Otevřený projekt zabezpečení webových aplikací OWASP (Open Web Application Security Project) identifikoval následující chyby zabezpečení jako 9 nejdůležitějších zranitelností [14]:

1. **Nesprávné webové rozhraní**
   Mnoho z těchto zařízení má vestavěný webový server, který hostí webovou aplikaci pro správu zařízení. Stejně jako kterémkoli webovém serveru, mohou být i zde v kódu chyby, které umožňují napadení zařízení.
2. **Neúčinná autentizace/autorizace**
   Zatímco často existují nedostatky v implementaci mechanismů ověřování/autorizace, větší problém je nevyužívání těchto funkcí, i když jsou danou technologií poskytovány. Spousta uživatelů připojí svá zařízení, nastaví nezbytnou konfiguraci pro funkci samotného zařízení, ale o zabezpečení a o dění komunikace „na vzduchu“ se již nikdo nezajímá.
3. **Nezabezpečené síťové služby**
   Přístroje IoT mohou mít služby pro diagnostiku a testování. Mohou dokonce mít služby ladění. Pokud se tyto služby nacházejí na otevřených, nejistých nebo nezabezpečených portech, představují potenciální bezpečnostní riziko.

4. **Nedostatečné šifrování přenosu**
   Pokud zařízení odesílá data prostřednictvím nejistého protokolu, může někdo tato data zachytit/číst. Pro spoustu uživatelů není vždy jasné, jaké informace mohou sdílet zařízení IoT.
5. **Zásady ochrany soukromí**
   Spousta mobilních zařízení mají různé programové nádstavby, jak sdílet informace z mobilního zařízení např. s rodinou či s jinou skupinou. Při nedostatečné programové autorizaci a autentizaci uživatele, kterému takové zařízení náleží mohou být ohrožena uživatelova soukromá data potažmo data celé skupiny.
6. **Nesprávné cloudové rozhraní**
   Mnoho zařízení IoT se připojuje ke cloudu. Pokud mají zařízení rozhraní pro správu cloudů, představuje to další potenciální bezpečnostní slabost. Mnohdy je toto bezpečnostní riziko pouze v rukou správců systému daného cloudu, které spočívá v zabezpečení serveru, na kterém je cloud spuštěn.
7. **Nezabezpečené mobilní rozhraní**
   Každý uživatel v dnešní době hodlá používat svůj mobilní telefon pro velkou spoustu funkcí. Mnoho zařízení IoT má mobilní rozhraní. Výrobce drobných IoT zařízení ale zajímá hlavně prodej a o zabezpečení jednotlivých zařízení se nejeví takový zájem. I z hlediska financí není pro výrobce vytvoření důmyslného zabezpečení rentabilní.
8. **Nedostatečné bezpečnostní funkce**
   I když je uživatel mnohdy přesvědčován o funkcích zabezpečení a šifrování, tak je možné, že dané zařízení šifrovaná zabezpečená data vůbec neodesílá a odesílá je v čisté nezabezpečené formě. Tato data mohou být následně zachycena, přečtena, pozměněna nebo zneužita.
9. **Nebezpečný software/firmware**
   Bezpečnostní záplaty pro řešení zjištěných bezpečnostních rizik nemusí být vždy dílem výrobce, ale může se jednat o škodlivý kód. Tento kód může následně odesílat data z mobilního zařízení na vzdálený server. Data mohou obsahovat čísla kreditních karet apod.

4.4 Budoucí bezpečnostní opatření

Systém IoT se skládá ze tří různých vrstev (aplikační, síťová a fyzická), z nichž každá má zranitelnosti a bezpečnostní rizika. K řešení těchto rizik a k úspěšné ochraně systému IoT se v této části uvádí vícevrstvý bezpečnostní přístup, který by měl být strukturován tak, aby poskytoval optimální vrstvenou ochranu na každé vrstvě v systému IoT. Následující informace vycházejí ze zdroje [13].

4.4.1 Zabezpečení fyzické vrstvy Internetu věcí

1. **Bezpečné zavádění**
   Autentizace a integrita softwaru v zařízení by měla být ověřena kryptografickým hashovacím algoritmem, který by poskytoval digitální podpisy. Nicméně kvůli nízkému zpracovatelskému výkonu na většině zařízení a jejich potřebě velmi nízké spotřeby energie nelze použít většinu kryptografických hashovacích algoritmů, kromě několika kryptografických hash funkcí, které jsou optimální pro zařízení s velmi nízkou spotřebou energie.
2. **Autentizace zařízení**
   Ve chvíli, kdy je do sítě zavedeno nové zařízení, je vhodné před příjmem nebo přenosem dat ověřit, zda je toto zařízení správně identifikováno a autorizováno.
3. **Integrita dat**
   V každém zařízení by měly být k dispozici mechanismy pro zjišťování chyb, aby nedocházelo k žádné manipulaci s citlivými údaji. Jsou upřednostňovány mechanismy nízké spotřeby energie, jako jsou cyklické redundantní kontroly CRC (Cyclic Redundancy Check), kontrolní součet a paritní bit, ale pro zajištění lepší detekce chyb by měla být aplikována některá z hashovacích funkcí.
4. **Důvěrnost údajů**
   Veškerá data by měla být zašifrována na každém zařízení před přenosem dat, aby se zajistila důvěrnost. Avšak vzhledem k vysoké spotřebě energie nelze použít silné kryptografické šifrovací funkce jako například AES. Namísto toho např. RSA (iniciály autorů Rivest, Shamir, Adleman) má nižší spotřebu energie a méně zpracovatelského výkonu a může být úspěšně implementován na zařízeních fyzické vrstvy.

4.4.2 Zabezpečení síťové vrstvy Internetu věcí

1. **Soukromí údajů**
   Je důležité zabránit neoprávněným přístupům k senzorovým uzlům pomocí autentizačních mechanismů a šifrování komunikace mezi jednotlivými uzly.
2. **Bezpečnost směrování**
   V systému IoT je důležité vybrat vhodný adaptivní směrovací protokol. Ideální směrovací protokol by měl být schopen detekovat chyby na síti a individuálně vybírat nejlepší trasy pro komunikaci senzorů. Dále by měl protokol předcházet nežádoucí opakované komunikaci za účelem šetření spotřeby elektrické energie.
3. **Integrita dat**
   Využitím kryptografických hashovacích funkcí je potvrzena celistvost dat přijatých na druhém konci. V případě prokázání narušení údajů jsou zaváděny mechanismy pro opravu chyb, které umožňují zmírnit problém.

4. **Síťový Firewall**
   Je vhodné striktní nastavení pravidel síťového filtru tak, aby byly povoleny pouze potřebné síťové porty a adresy zařízení. Blokace navázání cizí nežádoucí komunikace, zahazování nežádoucích dat.

4.4.3 Zabezpečení aplikační vrstvy Internetu věcí

1. **Bezpečnost dat**
   Autenzizace šifrování a integrity jsou na této úrovni rozhodující mechanismy pro zajištění ochrany soukromí celého systému a ochrany proti krádeži dat. Zabraňuje neoprávněnému přístupu do systému a zajišťuje důvěrnost systémových dat.
2. **Seznamy řízení přístupu**
   Nastavení zásad a oprávnění (kdo a jak může přistupovat a ovládat systém IoT) je zásadním bodem pro zajištění soukromí dat a korektní provoz systému. ACL (Access Control List) seznamy mohou blokovat/povolovat příchozí/odchozí provoz a umožňují/lokují přístup k žádostem od různých uživatelů uvnitř nebo vně sítě.
3. **Zabezpečující software**
   Bezpečnostní software jako antivirový nebo anti-spyware je důležitý pro spolehlivost, bezpečnost, integritu a důvěrnost systému IoT. Aby byla zajištěna trvalá ochrana systému a jeho důvěryhodnost, musí být hodnocení rizik, detekce narušení, fyzická bezpečnost a správa důvěryhodnosti povinnou součástí ve všech vrstvách systému Internetu věcí.

5 ANALÝZA BEZPEČNOSTNÍCH RIZIK LORAWAN

Za účelem zjištění zranitelnosti v síti LoRaWAN by výzkumníci měli „myslet jako útočník“, což znamená zvažovat a analyzovat možné útoky a najít důvody těchto útoků. Tato kapitola uvádí metody použité k analýze útoků a slouží jako **obecné schéma** pro následující kapitolu, v níž jsou detailně popsány specifické útoky. Tato kapitola vychází ze zdroje [15].

5.1 Název útoku

5.1.1 Identifikace cílů útočníka

Útoky jsou definovány ve dvou aspektech:

1. **útoky, které ohrožují vlastnosti zabezpečení sítě**,
2. **útoky, které ohrožují aktiva zabezpečení sítě**.

Vlastnosti zabezpečení v síti jsou vždy popsány pomocí trojice CIA (Confidentiality, Integrity, and Availability). Trojice CIA, která zahrnuje důvěrnost, integritu a dostupnost, odráží nejdůležitější požadavky na bezpečnost informací. V takovém případě je důležité zvážit tuto trojici prvků kvůli cílenému útoku na síť LoRaWAN.

Bezpečnostní prostředky odrážejí nejdůležitější parametry v síti. Pokud jsou tyto prostředky ohroženy, může být ohrožena i bezpečnost sítě.

Aktivum	Primární (P) nebo Sekundární (S) aktivum	Důvěrnost	Integrita
NwkSKey	P	Ano	Ano
AppSKey	P	Ano	Ano
AppKey	S	Ano	Ano
DevNonce	S	Ne	Ano
AppNonce	S	Ano	Ano
FrmPayload	P	Ano	Ano
DevAddr	S	Ne	Ano
Fcnt	S	Ne	Ano
ACK	S	Ne	Ano
MAC příkazy	S	Ano	Ano

Tab. 5.1: Aktiva v LoRaWan síti

Tabulka 5.1 znázorňuje, které prostředky jsou chráněny v sítích LoRaWAN a klasifikuje důležitost jednotlivých aktiv LoRaWAN. Například NwkSKey je primární

aktivum, protože útok, který by mohl ohrozit NwkSKey, je schopen přímo ohrozit síť. V případě sekundárních aktiv, jako je AppKey, útočník získá pouze službu DevNonce a pro narušení chodu sítě je ještě zapotřebí více prostředků.

Některá aktiva v tabulce 5.1 jsou důvěrná, zatímco jiná nejsou. Například důvěrnost AppNonce je chráněna, zatímco DevNonce není. Důvodem je to, že k odvození relačních klíčů je vyžadováno použití obou těchto položek. Integrita veškerých aktiv je chráněna. Popis jednotlivých položek [15]:

- **NwkSKey**
  NwkSKey je klíč síťové relace a používá se na síťovém serveru ke kontrole podpisu zprávy. Generuje se během aktivace uzlu. Pokud je ohrožena důvěrnost NwkSKey, může třetí osoba použít NwkSkey k vygenerování vlastní zprávy v LoRaWAN síti. Pokud je integrita NwkSkey narušena na síťovém serveru nebo na koncovém zařízení, může dojít k ohrožení komunikační relace zapojených zařízení a náležící zprávy v takovéto relaci mohou být zahozeny.
- **AppSkey**
  AppSKey je klíč aplikační relace a používá se na aplikačním serveru k dešifrování zpráv. Generuje se během aktivace uzlu. V případě ohrožení důvěrnosti AppSKey bude útočník schopen dešifrovat všechny zprávy. Důvěrnost celé sítě LoRaWAN bude ohrožena. Pokud dojde k ohrožení integrity aplikace AppSKey, aplikační server nebo koncové zařízení nebudou schopny správně dešifrovat zprávy. Získané údaje nemohou být důvěryhodné.
- **AppKey**
  AppKey je klíč aplikace a je využíván aktivační metodou OTAA za účelem odvození dvojice klíčů AppSKey a NwkSKey. Před aktivací je AppKey přiřazen vlastníkem aplikace ke koncovému zařízení i serveru. Pokud je důvěrnost aplikace AppKey ohrožena, bude útočník schopen spustit útok typu „replay“ na žádost o připojení a bude potencionálně schopen připojit škodlivé zařízení k síti. Pokud je integrita aplikace AppKey ohrožena, koncové zařízení se nebude moci připojit k síti LoRaWAN přes OTAA.
- **DevNonce**
  DevNonce je položka generovaná koncovým zařízením. Při aktivaci OTAA bude tato položka přenášena vzduchem z koncového zařízení do brány a síťového serveru. Poté budou položky DevNonce a AppNonce zakódovány pomocí AppKey. DevNonce může být přenášeno v otevřeném textu, protože bez klíče aplikační relace AppSkey nemůže útočník dosáhnout žádných útoků. Integrita DevNonce by měla být chráněna. Bez ochrany integrity budou klíče relace generované koncovým zařízením a serverem odlišné. Komunikační relace bude neplatná.

- **AppNonce**
  AppNonce je určitá forma jedinečného identifikátoru od síťového serveru. Používá se k vytváření klíčů relace s nástroji DevNonce a AppKey. Pokud není důvěrnost AppNonce chráněna, může útočník po získání AppKey snadno vypočítat NwkSKey a AppSkey a narušit bezpečnost celé sítě. Pokud je ohrožena celistvost AppNonce, budou klíče relace generované koncovým zařízením a serverem odlišné. Komunikační relace bude neplatná.
- **FrmPayload**
  FrmPayload slouží k přenosu dat senzoru např. teploty. Pokud útočník ohrozí důvěrnost FrmPayload, údaje ze senzoru budou útočníkům známy a způsobí problémy s ochranou soukromí. Pokud útočník ohrozí integritu FrmPayload, data senzorů přijatá serverem budou neplatná a neměla by být důvěryhodná.
- **DevAddr**
  DevAddr je identifikátor adresy koncového zařízení. Je v otevřeném textu. Je-li narušena jeho celistvost, bude komunikace mezi koncovým zařízením a serverem přerušena.
- **FCnt**
  FCnt je hodnota čítače v obou koncových zařízeních a serveru. Je v otevřeném textu. Pokud je integrita FCnt ohrožena, bude pro server a koncové zařízení obtížné udržet synchronizaci. Také změna hodnoty čítače může útočníkům usnadnit dosažení „replay“ útoku.
- **ACK**
  ACK (Acknowledgement) je speciální parametr zprávy, který slouží k potvrzení přijatých zpráv. Je v otevřeném textu. Pokud je poškozena celistvost ACK, je možné, že zpráva ACK bude upravena na normální zprávu a funkce ACK bude zakázána.
- **MAC příkazy**
  MAC příkazy mohou být odeslány ve složkách FOpts nebo FrmPayload. Některé příkazy, jako nastavení parametrů rádia, by měly být důvěrné. V opačném případě může útočník zjistit nastavení těchto parametrů a podle toho se přizpůsobit. Integrita MAC příkazů by měla být také chráněna.

5.1.2 Definice útočníkových možností

V úvaze, jaké mohou být útočníkovy možnosti, je vhodné počítat i s tím, že útočník má plnou kontrolu nad celým systémem. V síti LoRaWAN lze uvažovat následující schopnosti útočníka:

- znalost sítě a zařízení LoRa,
- zachycení a posílání zpráv vzduchem,

- zpracovávat a ukládat data,
- šifrovat a dešifrovat, pokud jsou známé klíče,
- fyzický přístup.

5.1.3 Definice vlastností LoRaWAN sítě

Mimo bezpečnostní prvky má síť LoRaWAN také fyzické vlastnosti, které mohou být ovlivňujícími faktory pro útok útočníka.

- **Bezdrátový přenos**
  Bezdrátový přenos má povahu všesměrového vysílání, což útočníkům poskytuje více příležitostí k napadení sítě. V bezdrátových senzorových sítích je směrování paketů nespojované a nemůže být považováno za spolehlivé, pokud není použit žádný jiný bezpečnostní mechanismus.
- **Latence**
  U tříd zařízené A a B v síti LoRaWAN dochází při přenosu ke zpoždění v sestupném směru a problémy se synchronizací mohou být kritické, pokud jsou v sestupném směru přenášeny zprávy kritických událostí.
- **Omezená paměť a úložiště**
  V bezdrátových senzorových sítích je paměť koncových zařízení obvykle malá a úložný prostor také. V takovém případě by bezpečnostní mechanismus pro takovou síť měl být jednoduchý a velikost kódu by také měla být malá. Z jiného pohledu mohou být tato zařízení zranitelná vůči útokům, které spotřebovávají velké množství paměti.

5.1.4 Popis útoku

V této části je popis jednotlivých útoků, který dokazuje jakým způsobem je možné napadnout rizikové části protokolu. Hlavním cílem této práce je nalézt zranitelnosti protokolu LoRaWAN a ukázat, že tyto nedostatky skutečně existují. Je popsána a graficky znázorněna základní sestava k provedení útoku.

5.1.5 Shrnutí

V této části jsou uvedena shrnutí jednotlivých scénářů pro tři typy útoků na síť LoRaWAN. Tato část poskytuje jasný dojem, jak a do jaké míry mohou tyto útoky ovlivnit funkčnost dané sítě.

6 ÚTOKY NA LORAWAN SÍTĚ

V této kapitole jsou uvedeny tři útoky proti protokolu LoRaWAN. Tyto útoky jsou detailně prezentovány a analyzovány podle kroků analýzy zranitelnosti uvedených v předchozí kapitole. Tato kapitola vychází ze zdroje [15].

6.1 Replay útok na zařízení aktivovaná pomocí ABP

6.1.1 Cíle útoku

Tento útok je navržen tak, aby dosáhl spoofingu a DoS. Na straně serveru je cílem útoku dosažení spoofingu. Po útoku přijme škodlivě opakovanou zprávu z koncového zařízení útočníka a server se domnívá, že zpráva pochází z pracovního koncového zařízení. U zařízení pro koncové uživatele je cílem útoku dosažení DoS. Po útoku se na serveru nepřijme zpráva, kterou vysílá zařízení. Perioda DoS závisí na výběru opakované zprávy.

6.1.2 Schopnosti útočníka

K dosažení tohoto útoku je využito:

- znalosti formátu fyzického užitečného zatížení zpráv LoRaWAN,
- znalosti kmitočtového pásma bezdrátové komunikace cíleného koncového zařízení,
- zachycené komunikace LoRaWAN,
- zařízení pro odesílání zpráv LoRaWAN o určité frekvenci,
- schopnosti ukládat a číst zprávy LoRaWAN sítě založené na otevřeném textu.

6.1.3 Rizika protokolu

- Aktivační metoda ABP má bezpečnostní nedostatky. U ABP aktivovaných koncových zařízení jsou použity statické klíče, což znamená, že po resetování zůstávají klíče stejné, nemění se a jsou použity i pro další budoucí relace. Síťový server tedy může přijmout škodlivou zprávu, která vyhovuje následujícím požadavkům:
  - klíče relace jsou stejné jako má jedno akceptované koncové zařízení,
  - DevAddr je totožný jako má jedno akceptované koncové zařízení,
  - hodnota čítače je přijatelná.

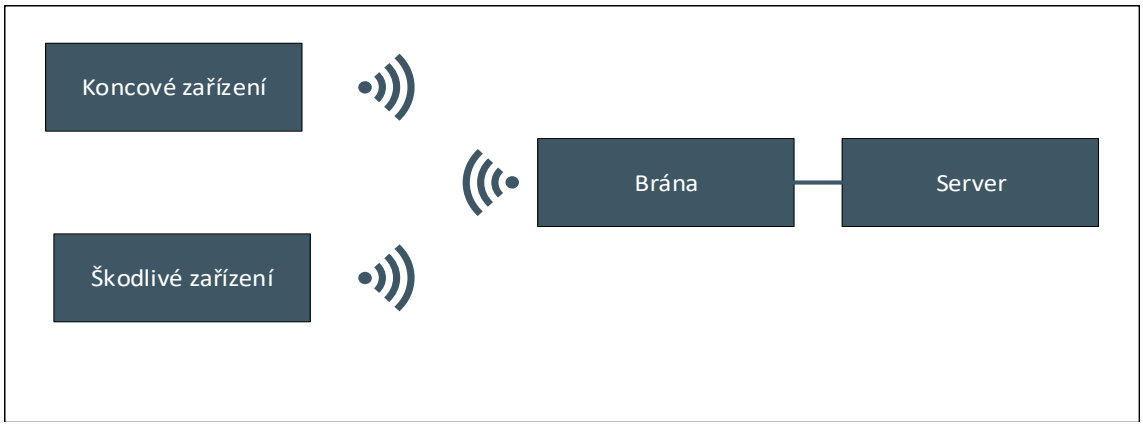
V takovém případě si útočník může před resetováním zvolit a znovu odeslat zprávy a server nemůže zjistit, zda jsou tyto zprávy z této relace nebo relace před obnovením.

- Čítače rámců se nepoužívají bezpečně. Ve specifikaci protokolu se tvrdí: „Po výměně zpráv JoinReq - JoinAccept nebo resetování koncového zařízení jsou čítače rámců na koncovém zařízení a na síťovém serveru pro toto koncové zařízení nastaveny na hodnotu 0.“ – Specifikace LoRaWAN 1.0.2. V tomto případě může útočník použít zprávy z poslední relace s většími hodnotami čítače a znovu je replikovat v aktuální relaci. Bez ohledu na to, zda je koncový přístroj aktivován systémem ABP nebo OTAA, je možné provést replay útok. Kromě manuálního resetování čítačů na obou stranách je dalším způsobem resetování „přetečení čítače“. Jakmile hodnota čítače dosáhne své maximální hodnoty, počítadlo se vynuluje a restartuje od 0. S hodnotami čítačů z poslední relace a se stejnými klíči relace může útočník také replikovat předchozí zprávy, aby odpojil komunikaci mezi koncovým zařízením a serverem. Hlavním bodem dosažení replay útoku je tedy zopakování hodnoty čítače. Z hlediska prosperity útoku má největší význam počkat na co možná nejvyšší hodnotu čítače a tuto hodnotu replikovat v nové relaci. Tím je dosaženo faktu, že veškeré zprávy s nižší hodnotou čítače budou ignorovány.

6.1.4 Popis útoku

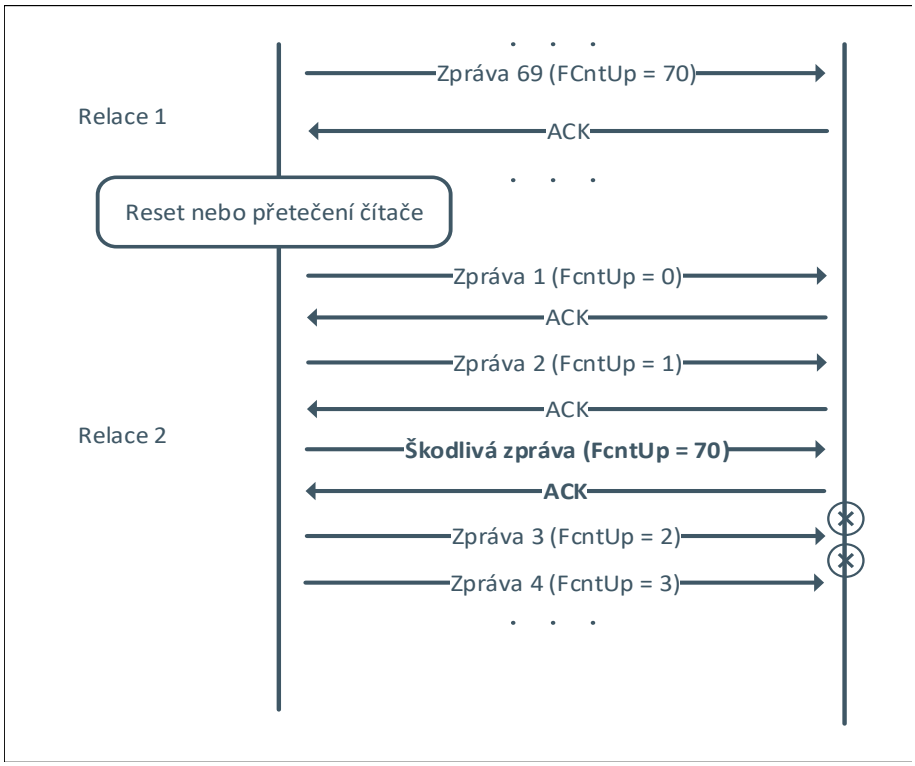
Obrázek 6.1 zobrazuje základní komponenty pro provedení replay útoku. Má-li být útok proveden, musí být dodrženy tyto kroky:

- zachycení zpráv – použití zařízení k zachycení zpráv vzestupného spojení na uzlu, který je aktivovaný metodou ABP,
- získání hodnoty FCntUp (hodnota čítače zařízení při vzestupném spojení),
- vyčkání na resetování zařízení nebo přetečení čítače,
- nalezení vhodné zprávy – výběr zachycené zprávy s vhodnou hodnotou čítače z databáze útočníka,
- replay – opětovné odeslání zprávy na bránu.



Obr. 6.1: Základní komponenty pro realizaci Replay útoku

Obrázek 6.2 znázorňuje příklad replay útoku. Maximální hodnota čítače je 16384. Škodlivou zprávou je zpráva zachycená v poslední relaci. Tato zpráva má se stejnou adresu zařízení, stejné klíče relace a větší hodnotou čítače. Pokud útočník pošle tuto zprávu do nové relace na síťový server a pokud je zpráva přijata, zprávy od postiženého koncového zařízení s menší hodnotou čítače jsou ignorovány.



Obr. 6.2: Příklad Replay útoku na uzly aktivované metodou ABP

6.1.5 Shrnutí

V případě uvedeného útoku může útočník využít programů pro zachytávání komunikace síťového provozu LoRaWAN a pomocí vysílače LoRa lze provést replay útok. Tento útok může být ve velké síti LoRaWAN mimořádně škodlivý pro koncová zařízení aktivovaná pomocí ABP. V malé síti LoRaWAN s pouze několika koncovými zařízeními může útočník pravděpodobně dlouho čekat na přetečení čítače. Ve velké síti LoRaWAN s násobným počtem koncových zařízení je však velmi snížena doba čekání na přetečení některého z koncových zařízení. Jakmile útočník dostane jednu největší možnou hodnotu čítače pro jedno koncové zařízení, může tuto zprávu pravidelně opakovat a trvale odepřít službu koncového zařízení. Pokud nejsou změněny klíče relace koncového zařízení, nemůže zařízení znovu fungovat. Pokud útočník nalezne způsob, jak zařízení resetovat (např. výpadek napájení), není nutné, aby čekal na přetečení čítače. Resetováním koncového zařízení a provedením replay útoku pomocí zprávy s největší hodnotou čítače, jsou zprávy od postiženého koncového zařízení ignorovány.

6.2 Útok Bit flipping

6.2.1 Cíle útoku

V českém jazyce tento útok znamená přehození jednotlivých bitů ve zprávě. Cílem útoku je prokázat, že integrita mezi síťovým serverem a aplikačním serverem není chráněna. Pokud má útočník možnost zachytit přenos, tak neexistuje způsob, jak může aplikační server rozpoznat, zda zpráva pochází od útočníka nebo ze síťového serveru.

6.2.2 Schopnosti útočníka

K dosažení tohoto útoku je využito:

- realizace útoku typu „muž uprostřed“ mezi síťovým serverem a aplikací,
- základních znalostí o fyzickém formátu užitečného zatížení,
- základních znalostí o typu zprávy od koncového zařízení.

Aby se zvýšila přesnost výsledků dešifrování, je vhodné mít také možnost vynulovat koncové zařízení.

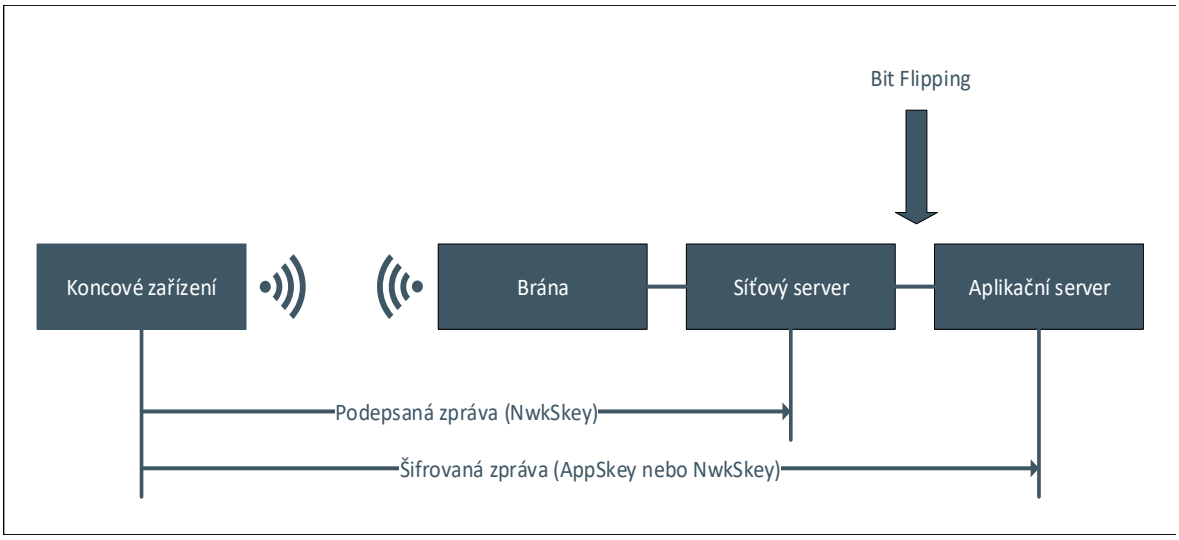
6.2.3 Rizika protokolu

Integrita mezi aplikačním serverem a síťovým serverem není kontrolována. Vzestupná spojení zprávy jsou zašifrovány a pak podepsány. Po přijetí síťovým serve-

rem použije síťový server NwkSKey ke kontrole podpisu zprávy. Poté jsou šifrované zprávy přijaty na síťovém serveru a poté zpracovány na aplikačním serveru. Mezi síťovým a aplikačním serverem mohou být údaje během manipulace změněny, protože při příchodu zpráv na aplikační server již není kontrolována celistvost šifrovaného textu.

6.2.4 Popis útoku

Obrázek 6.3 zobrazuje schéma zapojení v síti LoRaWAN pro realizaci bit flipping útoku. Útok může být proveden následovně:



Obr. 6.3: Sestava sítě pro realizaci bit flipping útoku

- Pokud má útočník přístup k síťovému serveru nebo je schopný přistupovat ke komunikaci mezi síťovým a aplikačním serverem, tak potencionálně schopen provést bit flipping útok.
- Jak popisuje výraz 6.1, je známo, že:

$$\begin{aligned} otevřený\ text \oplus klíč &= šifrovaný\ text \\ šifrovaný\ text \oplus klíč &= otevřený\ text \end{aligned} \tag{6.1}$$

- Pozice otevřeného textu odpovídá stejné pozici šifrového textu. Útočník cílí na úpravu šifrovaného text, aby ovlivnil otevřený text.
- Hlavičku, informace o směrování i příkazy lze také změnit.

6.2.5 Shrnutí

Tento útok je založen na předpokladu, že útočník může kompromitovat komunikaci mezi síťovým a aplikačním serverem. V síti LoRaWAN může být komunikačním připojením mezi síťovým a aplikačním serverem Ethernet, WiFi, 3G atd. Pokud je útočník schopen vytvořit útok typu „Muž uprostřed“, může později upravovat jakékoliv zprávy LoRaWAN. Pokud je modifikována hodnota pole FrmPayload, jsou aplikační data přijatá serverem chybná. Pokud je změněn parametr zprávy DevAddr, aplikační server se domnívá, že data pochází z jiného koncového zařízení. Je-li hodnota čítače změněna a zpráva splňuje podmínky, že hodnota čítače zprávy je menší než hodnota čítače v serveru, aplikační server odmítne a vyřadí zprávu.

6.3 Útok ACK Spoofing

6.3.1 Cíle útoku

Tento útok je navržen tak, aby poukázal na chybný návrh zprávy ACK v LoRaWAN síti. Cílem útoku je kompromitování zprávy ACK. Toho je docíleno tak, že senzor po odeslání zprávy na vzestupné spojení očekává potvrzovací zprávu ACK od síťového serveru na sestupném spojení. Pokud je však ACK zpráva ze serveru zachycena, tak je možné ji po modifikaci odeslat na sestupné spojení již kompromitovanou bránou. Senzor se potom domnívá, že obdržel ACK zprávu od serveru, nicméně komunikace mezi senzorem a serverem již reálně neexistuje.

6.3.2 Schopnosti útočníka

K dosažení tohoto útoku je využito:

- kontroly nad bránou,
- znalosti rozpoznání ACK zprávy a schopnosti znemožnit přenos ACK zpráv sestupného spojení z brány na koncové zařízení,
- schopnosti číst zprávy ACK a vhodně volit ACK s potřebnou hodnotou DevAddr a Fcnt,
- schopnosti odesílat vybrané zprávy ACK z brány na koncové zařízení.

6.3.3 Rizika protokolu

Ve většině případů je brána připojena k Internetu, čímž je systém LoRaWAN zranitelnější. Je zde možnost kompromitovat bránu a vytvořit z ní škodlivé zařízení. Chyba protokolu spočívá v tom, že zpráva ACK pro zprávu vzestupného spojení nedefinuje, která zpráva se skutečně potvrdila. Potvrzuje pouze poslední zprávu, kterou

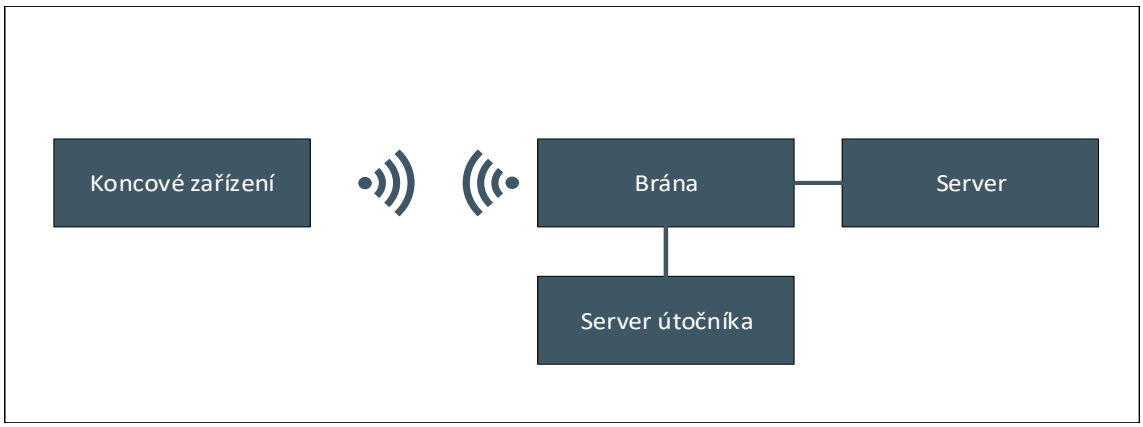
obdrží. Takže je možné, že kompromitovaná brána dokáže uchovat ACK potvrzení a použít ho pro budoucí zprávy.

6.3.4 Popis útoku

Pro tento útok existují dvě podmínky:

- ACK by mělo splňovat požadavky čítače. V ACK zprávě existuje parametr Fcnt, a aby bylo ACK akceptováno koncovým zařízením, tak by hodnota Fcnt v sestupném směru měla být větší než hodnota dnctr v koncovém zařízení. V tomto případě nelze ACK přehrát. Je také nutné zajistit, aby stejná ACK nebyla předtím přijata koncovým zařízením.
- Brána je kompromitována a funguje jako škodlivé zařízení.

Základní sestava útoku je zobrazena na obrázku 6.4. V této situaci už se počítá se skutečností, že brána je již ohrožena a útočník má nad ní plnou kontrolu.



Obr. 6.4: Sestava sítě pro realizaci ACK spoofing útoku

6.3.5 Shrnutí

Tento útok je založen na předpokladu, že brána je kompromitována škodlivým kódem. V tomto případě má útočník plnou kontrolu nad branou a může dosáhnout ACK spoofingu. Teoreticky má brána LoRaWAN pouze funkci přenosu zpráv. Pokud má útočník plnou kontrolu nad branou, může fyzicky poškozovat funkčnost sítě LoRaWAN.

7 MODUL IC880A-SPI

Cílem této kapitoly je poskytnout popis produktu včetně rozhraní, vlastností a výkonu modulu koncentrátoru iC880A-USB/SPI. Kapitola vychází ze zdroje [16].

7.1 Úvodní informace

Modul koncentrátoru iC880A je určen pro širokou škálu aplikací, jako jsou aplikace chytrého měření, IoT a M2M. Jedná se o vícekanálový vysoce výkonný modul vysílače/přijímače, který je navržen tak, aby přijímal současně několik paketů LoRa prostřednictvím různých faktorů rozprostřeného spektra na více kanálech. Modul koncentrátoru iC880A může být integrován do brány. Poskytuje robustní komunikaci mezi LoRa branou a velkým množstvím koncových uzlů LoRa, rozložených v širokém rozmezí vzdáleností. Systém iC880A potřebuje hostitelský systém pro správné fungování. Tento hostitelský systém může být počítač, který bude připojen k iC880A přes USB (Universal Serial Bus) nebo SPI (Serial Peripheral Interface Bus). iC880A je schopen přijímat až 8 LoRa paketů současně odeslaných s různými faktory rozprostřeného spektra na různých kanálech. Tato jedinečná schopnost umožňuje implementovat inovativní síťové architektury, které jsou výhodnější než jiné systémy s krátkým dosahem:

- Koncová zařízení (např:. čidla) mohou náhodně měnit kmitočet v každém přenosu. To poskytuje obrovské zlepšení robustnosti systému, pokud jde o odolnost proti rušení a rozmanitost rádiových kanálů.
- Koncová zařízení mohou dynamicky provádět přizpůsobení rychlosti připojení bez nutnosti složitosti protokolu. Neexistuje potřeba udržovat tabulku, která koncový bod používá konkrétní datovou rychlost, protože všechna data jsou demodulována paralelně.
- Kapacita na vzduchu může být zvýšena kvůli ortogonálním rozptylovým faktorům.
- Vzhledem k vysokému rozmezí lze použít hvězdicovou topologii. Výsledkem je jednoduchá implementace, která zabraňuje složitým síťovým vrstvám, bezdrátovým směrovačům a dalším provozům síťových protokolů.

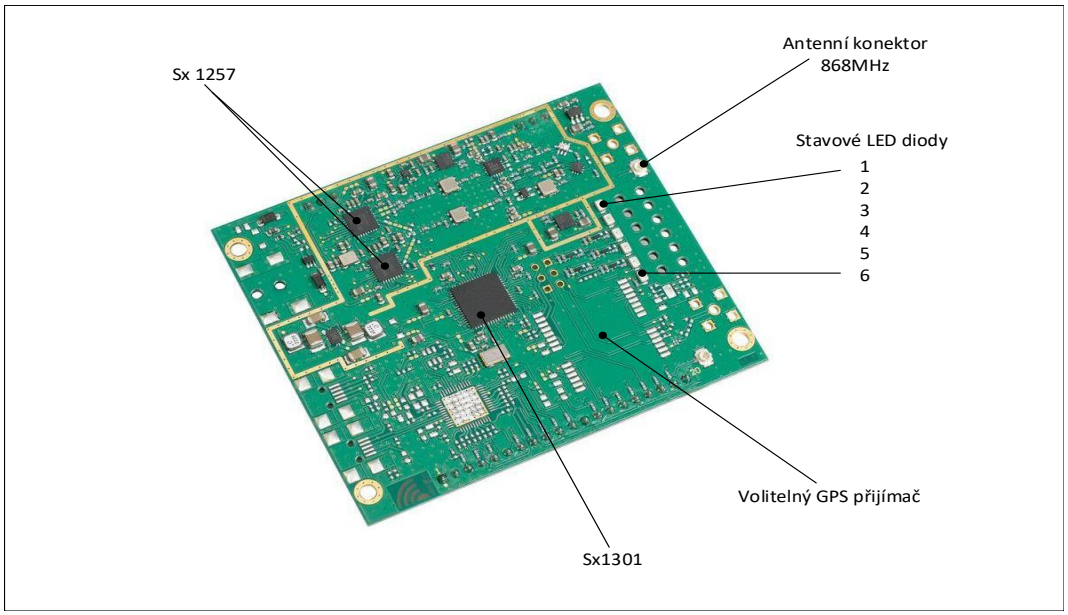
7.2 Vlastnosti modulu

- kompaktní velikost 79,8 x 67,3 mm,
- frekvenční pásmo 868 MHz,
- citlivost až -138 dBm,
- rozhraní USB nebo rozhraní SPI,

- základní procesor SX1301,
- 10 paralelních demodulačních cest,
- 1 (G) FSK demodulátor,
- 2 x SX1257 Tx/Rx na straně klienta[2],
- napájecí napětí 5 V,
- RF rozhraní optimalizované na 50Ω,
- výstupní výkon až 20 dBm,
- přijímač GPS (volitelné),
- rozsah do 15 km v přímem pohledu,
- rozsah několika km v městských oblastech,
- stavové diody LED (Light-Emitting Diode).

7.3 Popis modulu

Modul koncentrátoru je v současné době k dispozici ve dvou verzích, „iC880A-USB“ a „iC880A-SPI“, který je na obrázku 7.1. Je plánována budoucí verze s integrovaným GPS přijímačem.

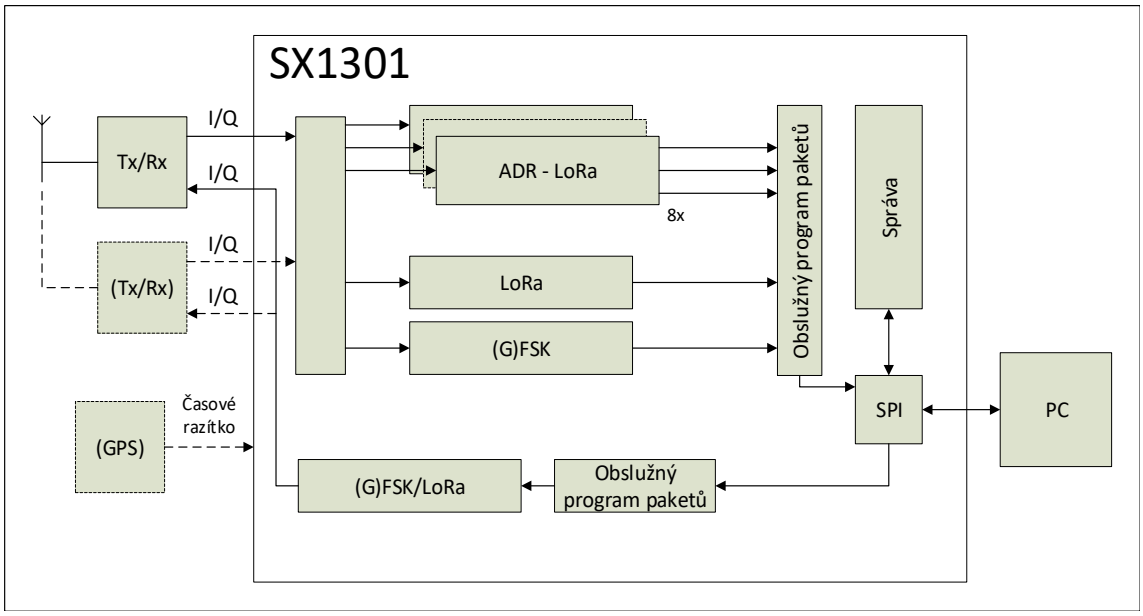


Obr. 7.1: Modul iC880A-SPI

[2]Tx jsou data vysílaná a Rx jsou data přijímaná.

7.3.1 SX1301

Model iC880A zahrnuje procesor SX1301 značky Semtech, který je určený pro zpracování digitálních signálů. Je speciálně navržen tak, aby nabízel průlomové brány v pásmech ISM po celém světě. Integruje koncentrátor LoRa IP. Na obrázku 7.2 je zobrazen zjednodušený blokový diagram.



Obr. 7.2: Blokový diagram procesoru SX1301

SX1301 je inteligentní základní pásmový procesor pro komunikaci v ISM s dlouhým dosahem. V přijímací části přijímá I a Q digitalizovaný proud bitů pro jeden nebo dva přijímače (SX1257). Demoduluje tyto signály, upravuje nastavení demodulátorů na přijatý signál a ukládá přijaté demodulované pakety do FIFO (First In First Out). Tyto pakety mají být získány z hostitelského systému – PC (Personal Compuer). V části vysílače jsou pakety modulovány pomocí programovatelného (G)FSK/LoRa modulátoru a odesílány na jeden vysílač (SX1257). Přijaté pakety mohou být časově označeny pomocí vstupu GPS. SX1301 má interní řídící blok, který přijímá mikrokód z hostitelského systému. Mikrokód je poskytován firmou Semtech jako binární soubor, který se načte do SX1301 při zapnutí. Řízení systému SX1301 hostitelským systémem se provádí pomocí HAL (Hardware Abstraction Layer) neboli hardwarové abstrakční vrstvy. Zdrojový kód vrstvy je poskytován společností Semtech. Procesor SX1301 má implementováno několik důležitých kanálů:

- **IF8 LoRa kanál**
  Tento kanál je připojen k jednomu vysílači SX1257 s libovolnou mezistupňovou frekvencí v povoleném rozsahu. Tento kanál je pouze LoRa. Demodulační

šířka pásma může být konfigurována tak, aby byla 125, 250 nebo 500 kHz. Datová rychlost může být nakonfigurována na libovolné dostupné datové rychlosti LoRa (SF7 až SF12), ale na rozdíl od IF0 až IF7 bude demodulována pouze nakonfigurovaná rychlost dat. Tento kanál je určen k tomu, aby sloužil jako vysokorychlostní zpětná vazba na jiné brány nebo vybavení infrastruktury.

- **IF9 (G)FSK kanál**
  Kanál IF9 je připojen k demodulátoru GFSK (Gaussian Frequency-Shift Keying). Pomocí tohoto demodulátoru je možné demodulovat libovolný starší FSK nebo GFSK signál. Šířku pásma kanálu a bitovou rychlost lze nastavit.
- **IF0 to IF7 LoRa kanály**
  Tyto kanály jsou připojeny k jednomu čipu SX1257. Šířka pásma kanálu je 125 kHz a nemůže být modifikována ani konfigurována. Každá frekvence kanálu IF může být individuálně konfigurována. Na každém z těchto kanálů lze přijímat libovolnou datovou rychlost bez předchozí konfigurace. Několik paketů používajících různé přenosové rychlosti (různé rozptylovací faktory) může být demodulováno současně i na stejném kanálu. Tyto kanály mají být použity pro masivní asynchronní hvězdicovou síť 10 000 koncových zařízení. Každé koncové zařízení může používat náhodný kanál (mezi IF0 až IF7) a jinou rychlost přenosu dat. Senzory umístěné poblíž brány obvykle používají nejvyšší možnou přenosovou rychlost v pevné šířce pásma kanálu 125 kHz (např. 6 kbit/s), zatímco vzdálené snímače použijí nižší datovou rychlost až na 300 bit/s (minimální přenosová rychlost LoRa v kanál 125 kHz). Digitální čip SX1301 skenuje 8 kanálů (IF0 až IF7) pro preambule všech datových rychlostí za všech okolností. Čip je schopen současně demodulovat až 8 paketů. Je možná libovolná kombinace faktoru šíření a mezifrekvence až pro 8 paketů (např. jeden paket SF7 na IF0, jeden SF12 paket na IF7 a jeden SF9 paket na IF1 zároveň). SX1301 může současně detekovat preambule odpovídající všem datovým kmitočtům na všech kanálech IF0 až IF7. Nemůže však současně demodulovat více než 8 paketů. Důvodem je to, že architektura SX1301 odděluje úlohu detekce preambule a odebírání signálů od procesu demodulace. Počet současně demodulovaných paketů (v tomto případě 8) je systémový parametr, který může být nastaven libovolně na jiné hodnoty pro specifický obvod zákazníka. Unikátní multikanálová demodulační kapacita SF7 až SF12 s více daty a kanály IF0 až IF7 umožňuje implementovat inovativní síťové architektury.

8 SESTAVENÍ SÍTĚ LORAWAN

V této kapitole jsou uvedeny informace o instalaci, potřebná schémata k fyzickému sestavení brány. Brána je realizována pomocí dnes již velmi známe platformy Raspberry Pi. Dále je zde popsáno přiřazení LoRa brány k serveru, vytvoření aplikace a k ní přidružená čidla. Instalace systému na SD (Secure Digital) kartu je provedena pomocí systému Linux.

8.1 Instalace systému Raspbian Stretch

Příkaz 8.1 provede stažení souboru do místa, ze kterého byl příkaz spuštěn např. **/home/user/Downloads**.

Výpis 8.1: Stažení obrazu

```
wget https://downloads.raspberrypi.org/raspbian_latest
```

Příkazy 8.2 a 8.3 zobrazí datová úložiště v PC. Vypsání tohoto příkazu je pro uživatele důležité z toho důvodu, aby chybně nevybral například zařízení, na kterém má nainstalovaný vlastní systém. Z Příkazu 8.3 vyplývá, že SD karta je zařízení **/dev/sdb**.

Výpis 8.2: Výpis programu fdisk pro zobrazení potřebného zařízení

```
sudo fdisk -l
Disk /dev/sda: 232,9 GiB, 250059350016 bytes, 488397168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x379d35df

Device    Boot    Start      End   Sectors Size Id Type
/dev/sda1 *        2048  1026047  1024000 500M  7 HPFS/NTFS/exFAT
/dev/sda2       1026048 147867647 146841600 70G 7 HPFS/NTFS/exFAT
/dev/sda3     147867648 357582847 209715200 100G 6 FAT16
/dev/sda4     357584894 488396799 130811906 62,4G 5 Extended
/dev/sda5     357584896 386881535 29296640 14G 83 Linux
/dev/sda6     386883584 472389631 85506048 40,8G 83 Linux
/dev/sda7     472391680 488396799 16005120 7,6G 82 Linux swap /
   Solaris
```

Výpis 8.3: Pokračování výpisu programu fdisk

```
Disk /dev/sdb: 14,4 GiB, 15489564672 bytes, 30253056 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device     Boot Start     End  Sectors Size Id Type
/dev/sdb1  *     2048 30253055 30251008 14,4G c W95 FAT32 (LBA)
```

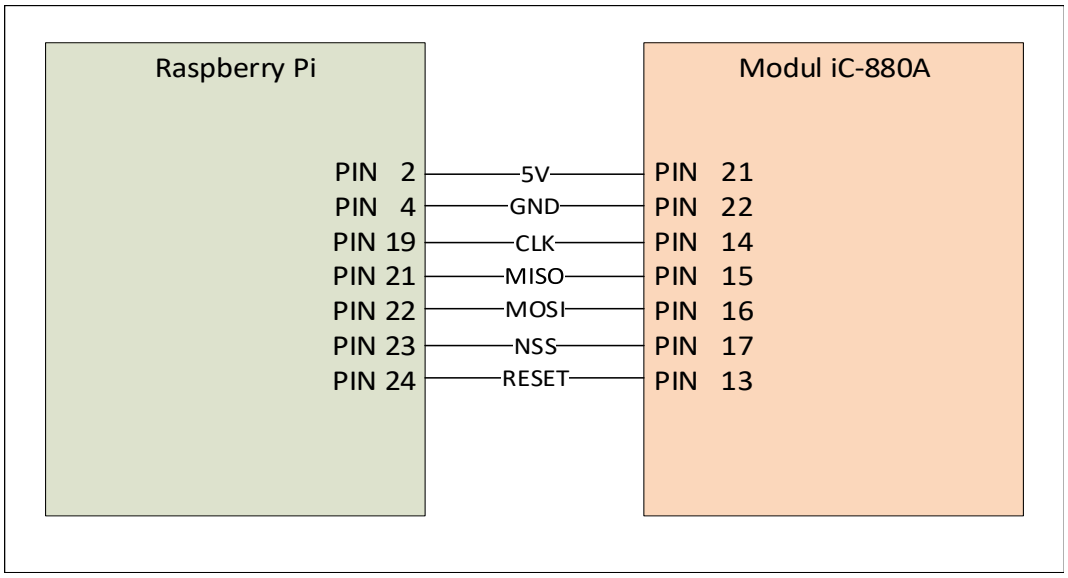
Příkaz 8.4 provede rozbalení archivu a následnou instalaci obsahu na zařízení **/dev/sdb**.

Výpis 8.4: Instalace na SD kartu

```
unzip -p 2017-09-07-raspbian-stretch.zip | sudo dd of=/dev/sdb
```

8.2 Schéma a fyzické provedení zapojení

Na obrázku 8.1 je možné vidět, jak je potřeba modul iC880A-SPI propojit s Raspberry Pi. Na obrázcích 8.2 a 8.3 pak fyzickou realizaci brány.



Obr. 8.1: Schéma zapojení Raspberry Pi s modulem iC880A-SPI

Obr. 8.2: Realizace brány propojením Raspberry Pi s modulem iC880A-SPI

Obr. 8.3: Realizace brány propojením Raspberry Pi s modulem iC880A-SPI

8.3 Konfigurace Raspberry Pi

Po zavedení systému Raspbian Stretch je provedeno systémové přihlášení pomocí standardního uživatelského účtu **pi**, který je pro platformu Raspberry Pi defaultní. Pomocí příkazu 8.5 je vytvořen druhý účet **ttn**, který nese název verze programu LoRa brány.

Výpis 8.5: Vytvoření uživatele ttn

```
sudo adduser ttn
```

Pomocí příkazu 8.6 přidáme k účtu uživatele **ttn** heslo **iotsc535**.

Výpis 8.6: Přiřazení hesla k účtu uživatele ttn

```
sudo passwd ttn
```

Po vytvoření uživatele je ještě potřeba přidělit mu potřebná práva, a sice práva **sudo**. Tato práva umožní jakémukoliv uživatelskému účtu spouštět nebo modifikovat soubory s právy uživatele **root**. Toho je docíleno modifikací souboru **sudoers** pomocí nějakého systémového editoru, např. **nano**. Znázorněno v příkazu 8.7.

Výpis 8.7: Modifikace souboru sudoers

```
sudo nano /etc/sudoers
```

Příkazy 8.8 a 8.9 znázorňují, jak přesně musí být soubor sudoers modifikován, aby práva sudo pro uživatele ttn fungovala.

Výpis 8.8: Výpis souboru sudoers

```
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead
   of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults       env_reset
Defaults       mail_badpass
Defaults       secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin
   :/usr/bin:/sbin:/bin"
```

Výpis 8.9: Výpis souboru sudoers

```
# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
ttn     ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

V tento okamžik je důležité odhlásit původní uživatelský účet „pi“. To je provedeno příkazem 8.10.

Výpis 8.10: Odhlášení úživatele „pi“

```
logout
```

Přihlášení pod novým uživatelským účtem „ttn“ se provede pomocí uživatelského jména **ttn** a hesla **iotsc535**, které bylo k účtu přiřazeno. Veškeré další příkazy jsou tedy prováděny právě pod uživatelským účtem **ttn**. Nyní je možné původní účet smazat. Tento krok není nezbytný, ale z hlediska administrace to umožní lepší přehlednost. Pokud by byl uživatel „pi“ v systému ponechán, je velmi důležité přiřadit tomuto účtu jiné než defaultní heslo. Defaultní přihlašovací údaje jsou totiž na webových stránkách systému Raspbian volně dostupné. Uživatelský účet „ttn“ bude hlavním účtem pro spouštění budoucího programu LoRa brány. Smazání původního uživatele se provede příkazem 8.11.

Výpis 8.11: Smazání uživatele „pi“

```
sudo userdel pi
```

Dalším krokem je nastavení síťového rozhraní. To bude prozatím nastaveno na dynamické přidělování adres. Pro pozdější využití, kdy bude brána v laboratoři, bude nastaveno adresování statické. Spuštění konfigurace provedeme příkazem 8.12.

Výpis 8.12: Modifikace souboru /etc/network/interfaces

```
sudo nano /etc/network/interfaces
```

Příkaz 8.13 již ukazuje dané nastavení síťového rozhranní.

Výpis 8.13: Nastavení souboru /etc/network/interfaces

```
# interfaces(5) file used by ifup(8) and ifdown(8)
# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

    auto eth0
    allow-hotplug eth0
    iface eth0 inet dhcp
```

Pro aplikaci nastavení síťového rozhraní, je nutné restartovat program, který síťování zajišťuje. To lze provést příkazem 8.14.

Výpis 8.14: Resetování síťování

```
sudo /etc/init.d/networking restart
```

Aby bylo možné provádět konfiguraci zařízení i vzdáleně, je nutné znát jeho IP adresu. Tu zjistíme pomocí příkazu 8.15.

Výpis 8.15: Výpis programu ifconfig

```
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
       inet 192.168.255.105 netmask 255.255.255.0 broadcast
          255.255.255.255
       inet6 fe80::d17e:e5d3:b1cd:8838 prefixlen 64 scopeid 0x20<
          link>
       ether b8:27:eb:b3:07:46 txqueuelen 1000 (Ethernet)
       RX packets 1637 bytes 171934 (167.9 KiB)
       RX errors 0 dropped 3 overruns 0 frame 0
       TX packets 1218 bytes 159734 (155.9 KiB)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

8.4 Spuštění LoRa brány

Po konfiguraci zařízení je potřeba nainstalovat program GIT, který slouží k získání externího repozitáře do místního systému. To umožní příkaz 8.16.

Výpis 8.16: Instalace programu GIT

```
sudo apt-get install git
```

Po instalaci programu GIT je nutné implementovat repozitář do systému pomocí příkazu 8.17. Tento příkaz vytvoří repozitář v adresáři, ve kterém se aktuálně nachází systémový uživatel. V tomto případě bude konkrétně v **/home/ttn/LoRa/ic880a-gateway**.

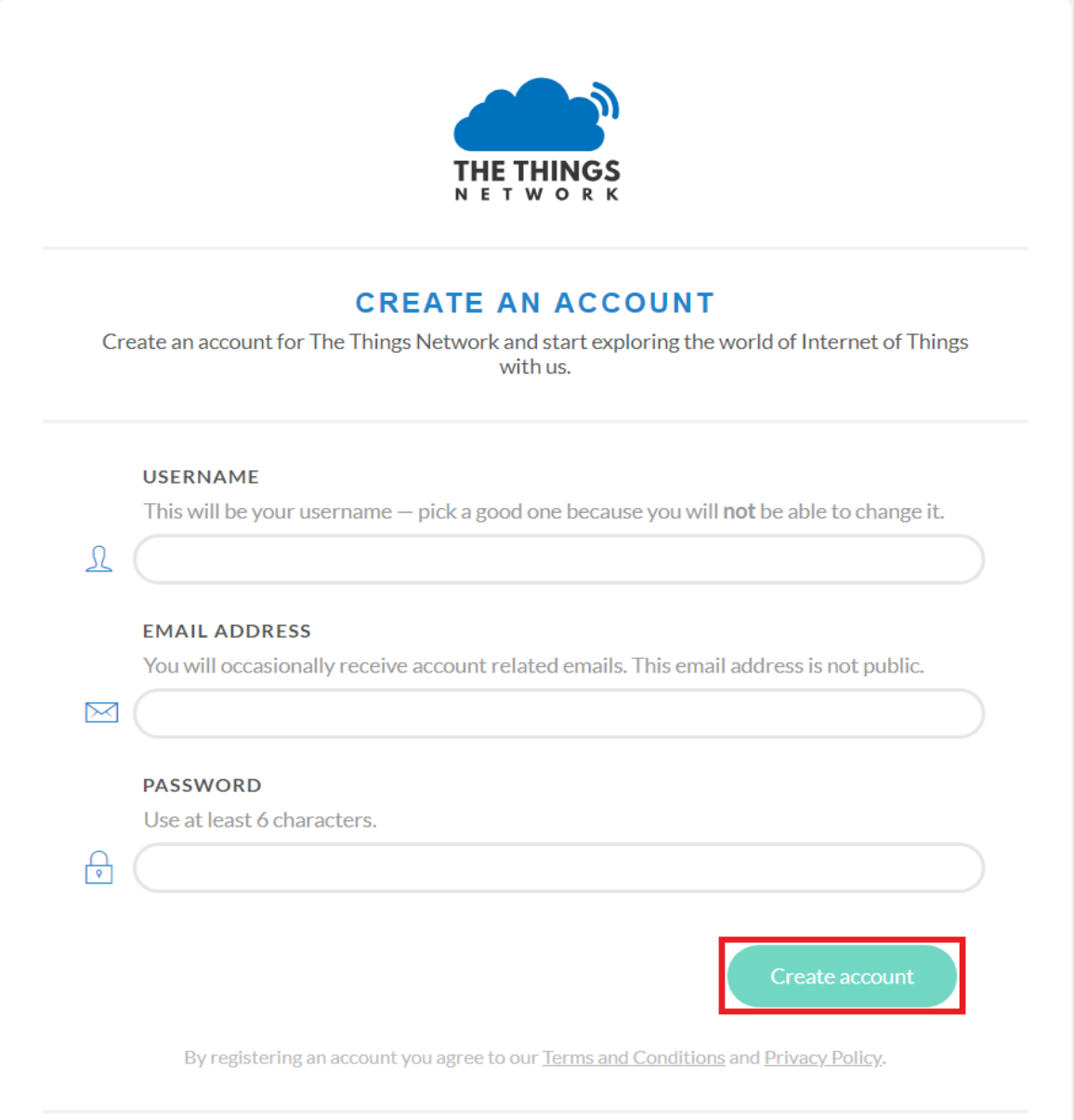
Výpis 8.17: Instalace programu GIT

```
git clone -b spi https://github.com/ttn-zh/ic880a-gateway.git
```

Nyní je potřeba vytvořit účet na některém ze síťových serverů. K tomu byla využita adresa `https://www.thethingsnetwork.org/`. „The things network“ je internetový portál, který implementuje síťový server systému LoRaWAN. Na adrese se zvolí tlačítko **SIGN UP**, jak ukazuje obrázek 8.4.

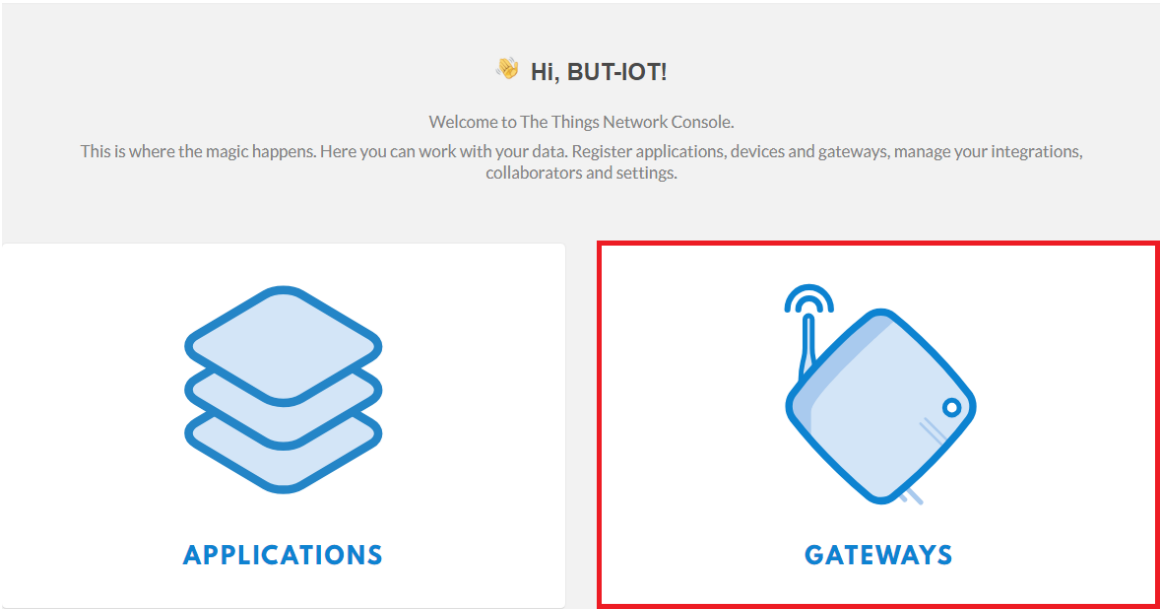
Obr. 8.4: Vytvoření účtu na portálu The Things Network

Po zvolení tlačítka se zobrazí formulář, který je potřeba vyplnit. Jako první údaj se vyplňuje položka **USERNAME** neboli uživatelské jméno – v tomto případě BUT-IOT (Brno University of Technology-Internet of Things). Druhá položka je kontaktní email a třetí položka je heslo. Formulář je znázorněn na obrázku 8.5. Údaje se potvrdí tlačítkem **Create account**, a tím je účet úspěšně vytvořen.



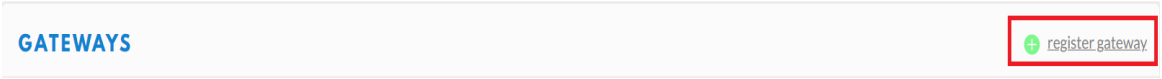
Obr. 8.5: Formulář registrace účtu na portálu The Things Network

Následující krok spočívá v registraci sestavené brány na síťovém serveru. K tomu je využita adresa `https://console.thethingsnetwork.org/`. Po otevření této adresy se zobrazí konzola systému ttn. Zde je zvolena možnost **GATEWAYS** pro přidání brány, jak ukazuje obrázek 8.6.



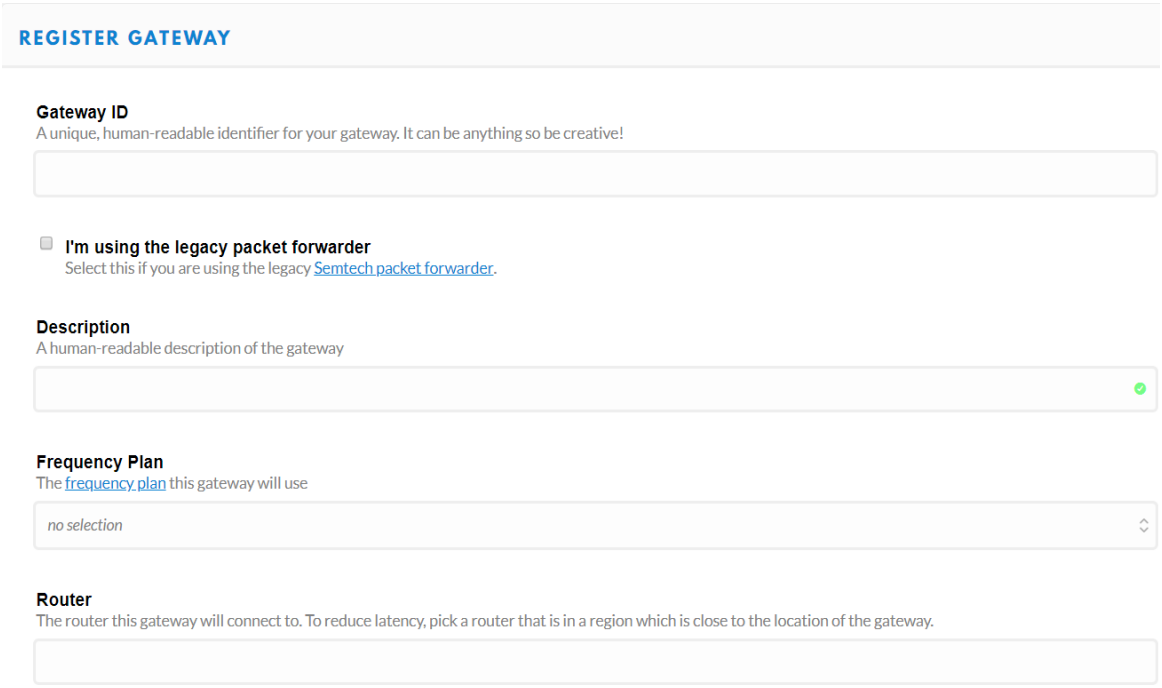
Obr. 8.6: Konzola systému ttn pro registraci brány

Po kliknutí na možnost brány je zapotřebí přidat bránu pomocí tlačítka **register gateway**, jak ukazuje obrázek 8.7.

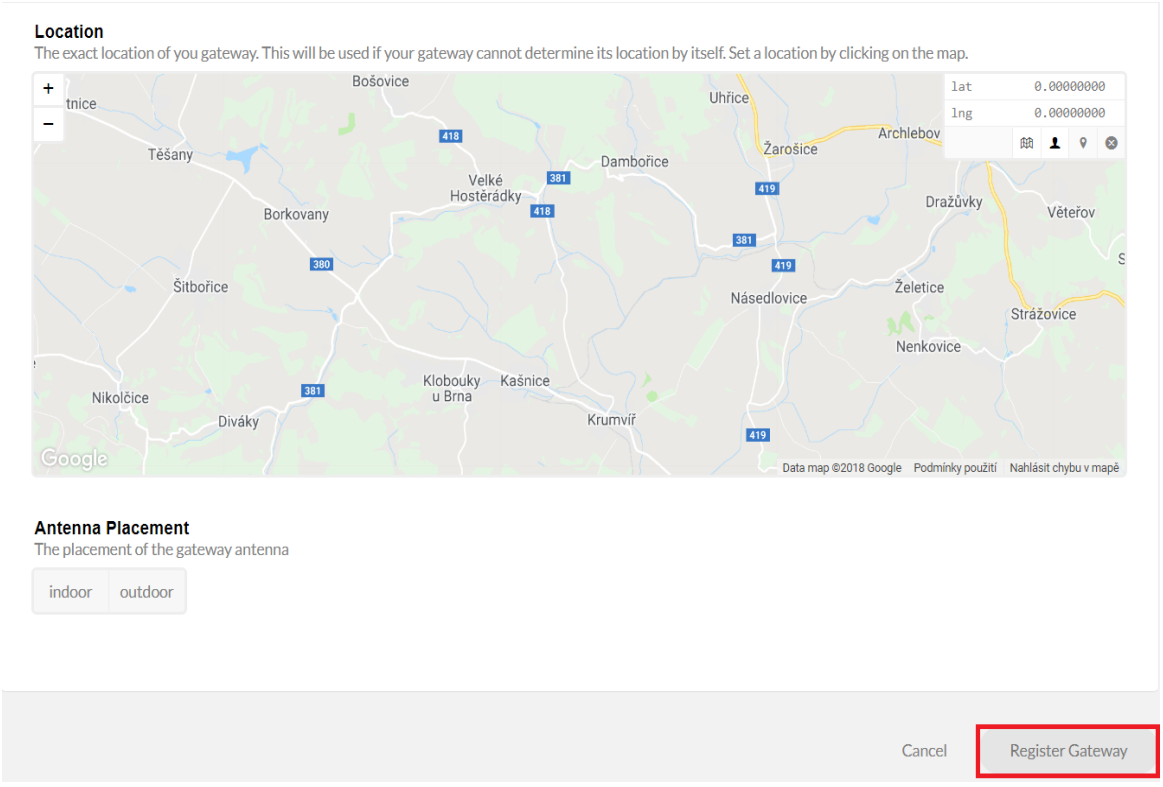


Obr. 8.7: Registraci brány

Po volbě registrace se objeví další formulář pro vyplnění potřebných údajů. Prvním údajem je **Gateway ID**, což je unikátní identifikátor brány. Tento identifikátor však slouží pouze uživateli pro lepší orientaci a administraci a může být volen jakkoliv. Dalším údajem je zaškrtávací možnost, která potvrzuje/vyvrací, že uživatel na dané bráně používá nativní paket forwarder. Dalším údajem je **Description**. Toto pole slouží k detailnějšímu popisu zařízení, nicméně není nutné ho vyplňovat. Další položkou je frekvenční plán **Frequency Plan**, který definuje, na jakém kmitočtu brána pracuje. Je zde také je volba směrovače **Router**. Posledními položkami formuláře jsou **Location** a **Antenna Placement**, které upřesňují polohu zařízení, popřípadě jestli je zařízení umístěno uvnitř nebo venku. Po vyplnění údajů se klikne na možnost **Register Gateway**, a tím se brána se všemi údaji vytvoří. Formulář je zobrazen na obrázcích 8.8 a 8.9.



Obr. 8.8: Formulář registrace brány



Obr. 8.9: Formulář registrace brány

V tento moment je však brána ještě stále nepřipojena, protože je potřeba nainstalovat software programu LoRa brány na zařízení Raspberry Pi. Toho je docíleno spuštěním instalačního skriptu v nainstalovaném repozitáři **/home/ttn/LoRa/ic880a-gateway** jak ukazuje příkaz 8.18.

Výpis 8.18: Instalace skriptu install.sh

```
sudo ./install.sh
```

Po provedení daného příkazu, se zobrazí výpis 8.19, který informuje, že na portu **eth0** byl detekován identifikátor brány. Je důležité si poznamenat tento identifikátor (B827EBFFFEB30746), protože bude použit ještě v lokálním konfiguračním souboru ve formátu json.

Výpis 8.19: Výpis skriptu install.sh

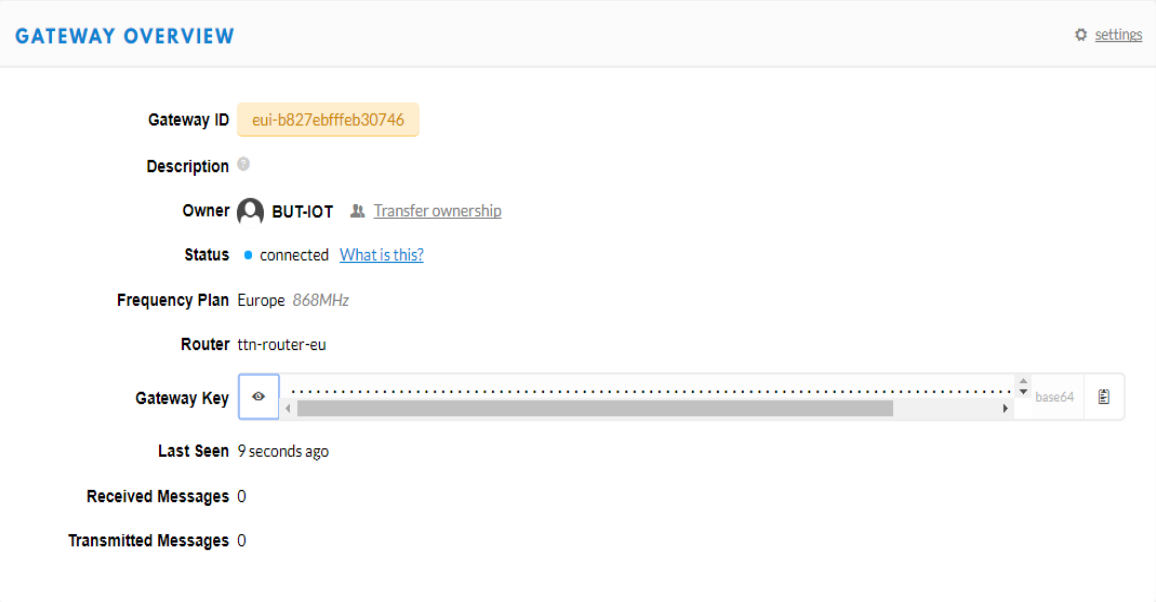
```
Version spi
Updating installer files...
Already up-to-date.
Gateway configuration:
Detected EUI B827EBFFFEB30746 from eth0
Do you want to use remote settings file? [y/N]
```

Po zjištění identifikátoru je potřeba ještě instalaci přerušit a tento identifikátor zapsat do souboru ve formátu **json** na internetovém portálu `https://github.com/ttn-zh/gateway-remote-config`. Soubor musí mít stejný název jako identifikátor brány, v tomto případě **B827EBFFFEB30746.json**. Jak je možné vidět ve výpisu 8.20, tak v souboru jsou informace o identifikátoru brány, adresa síťového serveru, port pro sestupné i vzestupné spojení, jsou zde také geografické informace v podobě zeměpisné šířky a délky, frekvenční plán pro Evropu tj. 868 MHz a směrovač, ke kterému se bude připojovat brána – v tomto případě **ttn-router-eu**. Je logické, že tyto informace korespondují s informacemi vyplněnými dříve ve formuláři registrace brány.

Výpis 8.20: Obsah souboru B827EBFFFEB30746.json

```
Gateway EUI is: B827EBFFFEB30746
{
  "gateway_conf": {
  "gateway_ID": "B827EBFFFEB30746",
  "servers": [
    {
      "server_address": "router.eu.thethings.network",
      "serv_port_up": 1700,
      "serv_port_down": 1700,
      "serv_enabled": true
    }
  ],
  "ref_latitude": 49.181646,
  "ref_longitude": 16.692134,
  "ref_altitude": 0,
  "contact_email": "",
  "description": "BUT-IOT"
  }
}
```

Nyní je nutné znovu spustit instalátor programu brány pomocí příkazu 8.18. Instalátor vyzve k potvrzení instalace a obsah celého repozitáře se nainstaluje do adresáře **/opt/ttn-gateway**. Po této operaci se zařízení automaticky restartuje. Po restartu se spustí program brány a také program paket forwarderu – v tomto případě **poly_pkt_fwd**, který je nativní pro ttn instalaci. Tento paket forwarder zapříčiní předávání LoRa paketů na server. Brána již je na serveru ve stavu **connected** jak ukazuje obrázek 8.10.



Obr. 8.10: Registrace brány do portálu The Things Network

8.5 Testy

Po kompletní instalaci je spuštěn test z adresáře **/opt/ttn-gateway/util-tx-test/**, který slouží k testování odesílaných paketů. Znázornění ve výpisu 8.21.

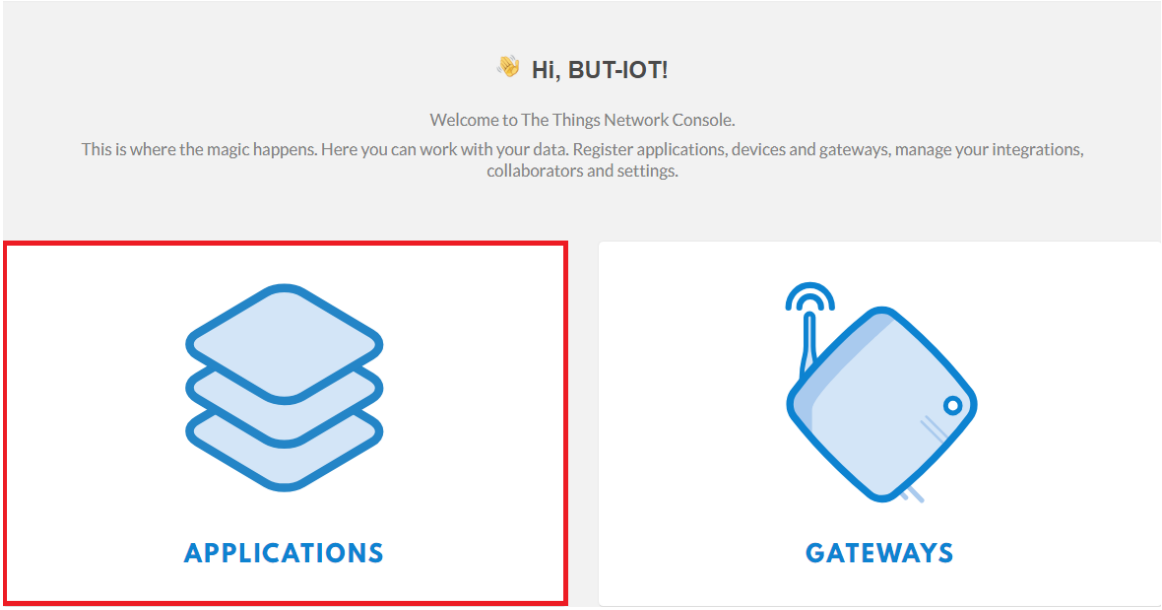
Výpis 8.21: Testování odesílaných paketů

```
sudo ./util_tx_test -f 868 -r 1257
Sending -1 packets on 868000000 Hz (BW 125 kHz, SF 10, CR 1, 16 bytes
    payload, 8 symbols preamble) at 14 dBm, with 1000 ms between
   each
INFO: concentrator started, packet can be sent
Sending packet number 1 ...OK
Sending packet number 2 ...OK
Sending packet number 3 ...OK
Sending packet number 4 ...OK
Sending packet number 5 ...OK
Sending packet number 6 ...OK
Sending packet number 7 ...OK
Sending packet number 8 ...OK
Sending packet number 9 ...OK
Sending packet number 10 ...OK
```

8.6 Vytvoření aplikace

Aby bylo možné ke kompletní LoRaWAN síti přidat nějaká čidla, je nejdříve nutné vytvořit tzv. aplikaci, ke které budou jednotlivá čidla přidružena. V praxi je aplikací myšleno například snímaní teploty. Do této aplikace jsou přidána všechna teplotní čidla, ze kterých jsou hromadně sdružovaná data. Tato data mohou být později využita k různým typům automatizace, či měření regulací.

Na adrese `https://console.thethingsnetwork.org/` je opět otevřena ttn konzola. Nyní je však zvolena položka **APPLICATIONS**, jak ukazuje obrázek 8.11.



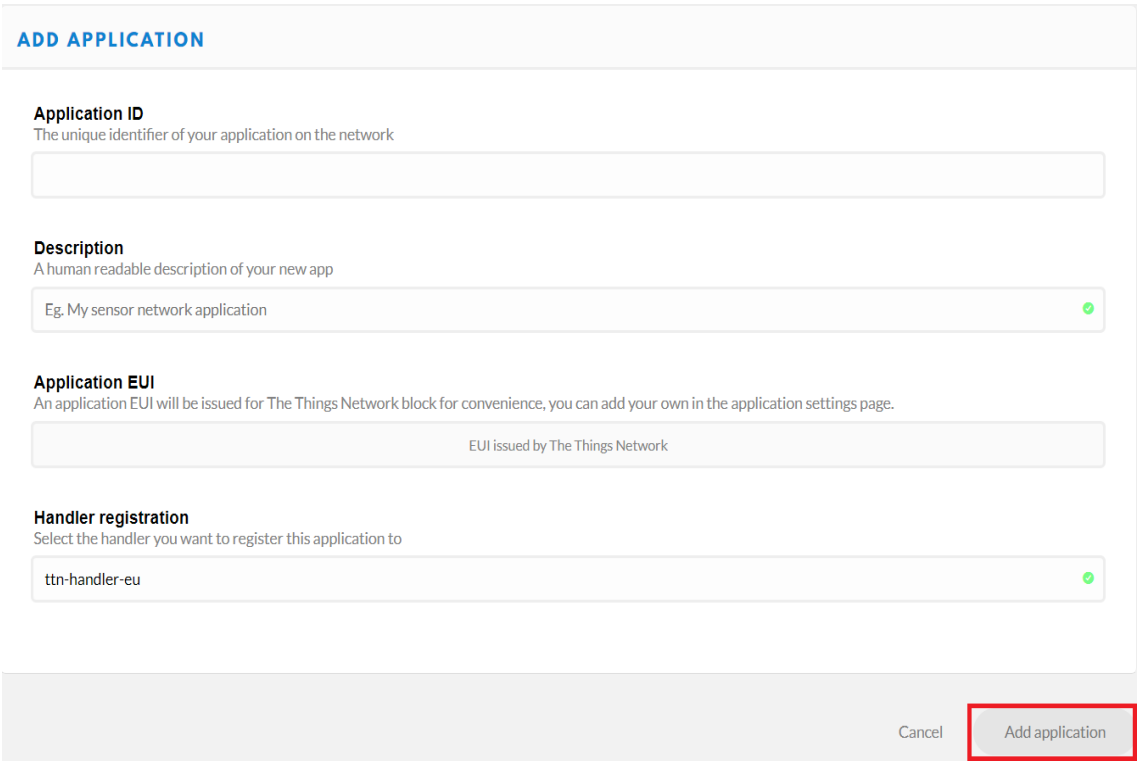
Obr. 8.11: Nastavení aplikace na portálu The Things Network

Po kliknutí na položku aplikace je zvolena možnost **add application** viz obrázek 8.12.

Obr. 8.12: Vytvoření aplikace na portálu The Things Network

Po této volbě se otevře formulář viz obrázek 8.13, kde je zapotřebí vyplnit identifikátor aplikace **Application ID** (lze pojmenovat jakkoliv), popis zařízení **Description** (volitelné), manipulátora aplikace **Handler registration**. Server při založení aplikace také vytváří identifikátor AppEUI, který reprezentuje globální identifikaci aplikace a slouží k pozdější aktivaci jednotlivých zařízení. Proces vytvoření aplikace

je po vyplnění údajů dokončen kliknutím na možnost **Add application**. Nyní je již možné k aplikaci přiřadit jednotlivá čidla.

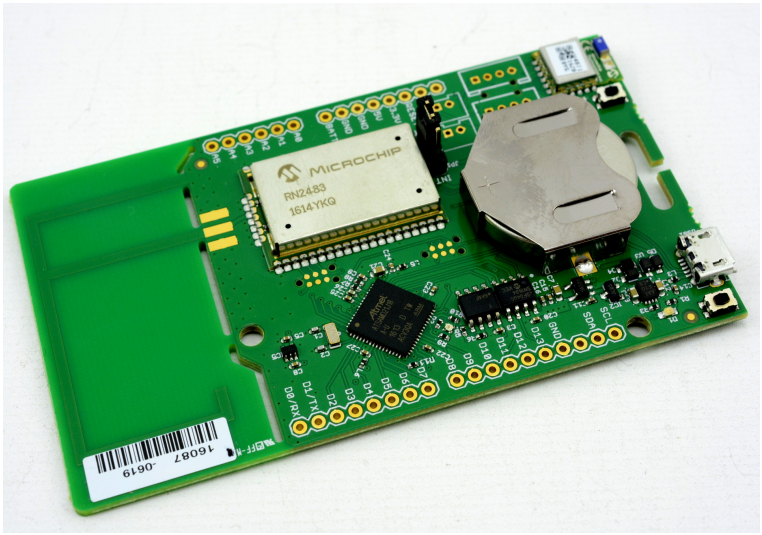


Obr. 8.13: Formulář vytvoření aplikace na portálu The Things Network

8.7 Konfigurace čidel

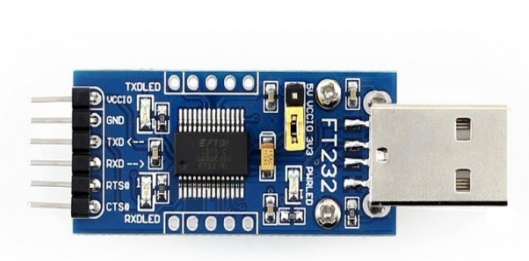
V práci byly použity dva typy čidel:

1. **Čidlo aktivované metodou OTAA, konkrétně SODAQ Explorer**
   V případě tohoto čidla se spíše jedná o jednodeskový programovatelný počítač, založený na mikrokontroléru ATmega od společnosti Atmel. Deska je programovatelná ve vývojovém prostředí Arduino IDE. V tomto případě byl ze stránek `http://support.sodaq.com/sodaq-one/lorawan/` stažen vzorový příklad a upraven tak, aby korespondoval s klíči, vygenerované serverem. Deska SODAQ Explorer na obrázku 8.14. Program je umístěn v příloze této práce.



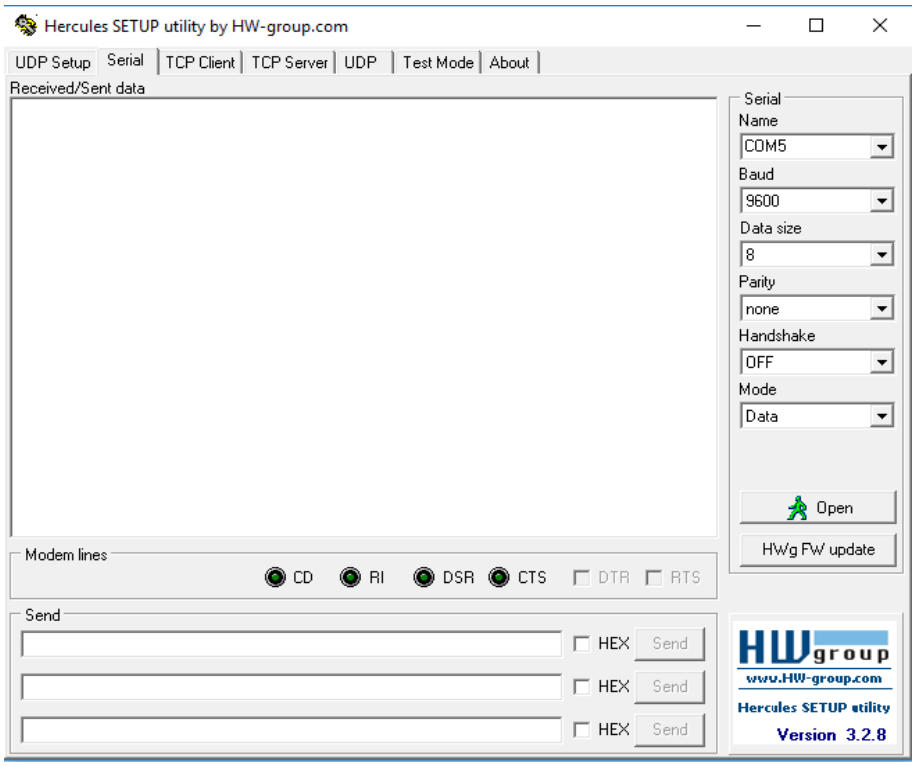
Obr. 8.14: Vývojová deska SODAQ Explorer

2. **Čidlo aktivované metodou ABP, konkrétně SolidusTECH mini UNI**
   Toto čidlo je před použitím nutné naprogramovat pomocí USB/UART (Universal Asynchronous Receiver and Transmitter) převodníku 8.15. Prvním krokem je nainstalování ovladačů pro daný FTDI (Future Technology Devices International) čip na převodníku. Ovladače jsou dostupné na webové adrese `http://www.ftdichip.com/Drivers/VCP.htm`. Převodník musí mít nastavené úrovně signálu a zejména napájení na 3,3 V jinak hrozí poškození čidla. Poté je potřeba zjistit, na jaký COM port (Windows) popřípadě tty (Linux) se převodník připojil. Nakonec se do čidla pomocí převodníku a příkazu v terminálu nahrají klíče a další parametry, které vygeneruje server. Čidlo SolidusTECH mini UNI na obrázku 8.16 a konfigurační terminál s patřičným nastavením na obrázku 8.17. Terminál je dostupný na adrese `https://hercules-setup.soft32.com/`.



Obr. 8.15: Převodník USB/UART

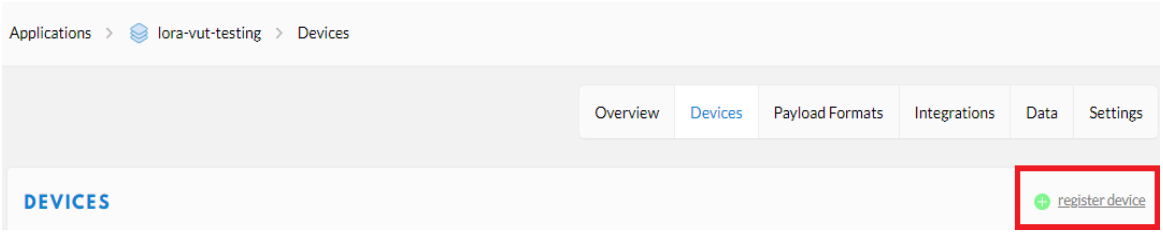
Obr. 8.16: Teplotní čidlo SolidusTECH mini UNI



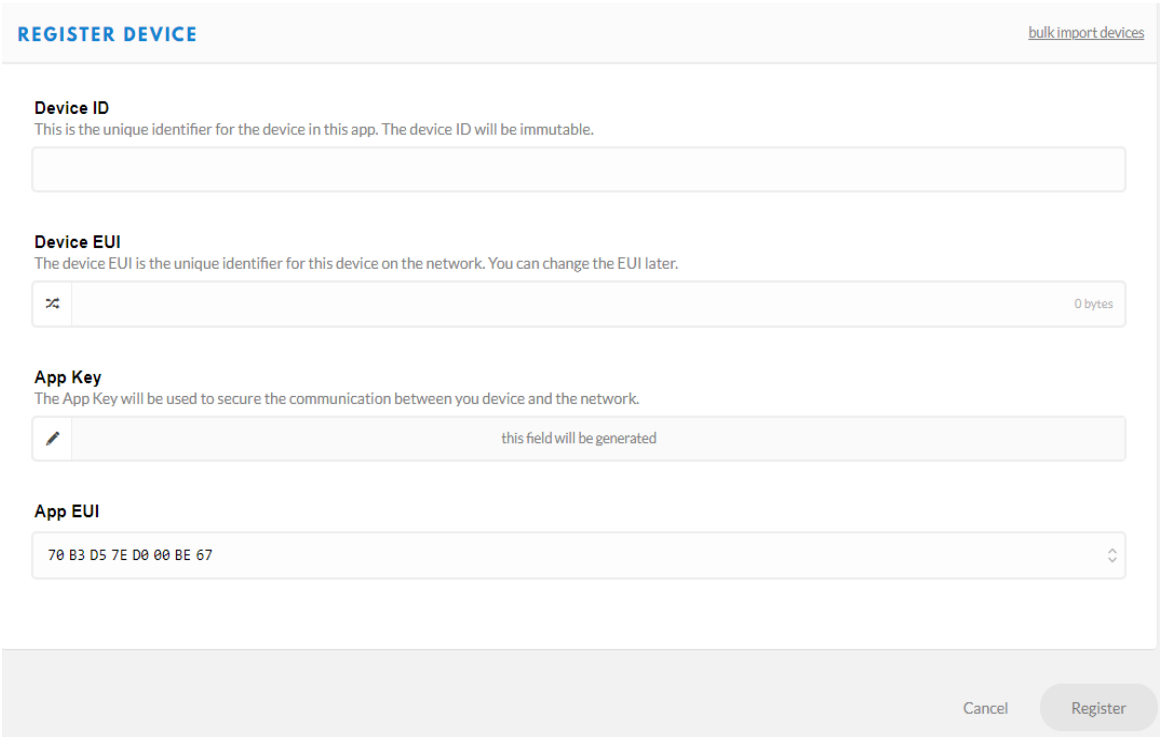
Obr. 8.17: Konfigurace čidla pomocí terminálu Hercules

8.8 Aktivace čidel v aplikaci

Posledním krokem pro zprovoznění LoRa komunikace je přiřazení čidel aplikaci a jejich následná aktivace. V sekci **Applications > lora-vut-testing > Devices** se nachází tlačítko **register device**, viz obrázek 8.18. Po stisku tlačítka se zobrazí formulář registrace zařízení viz obrázek 8.19. Zde se vyplňuje uživatelský identifikátor zařízení **Device ID**, poté položka textbfDevice EUI, což je řetězec znaků, který generuje server, nebo je uložen v zařízení již z výroby. Následuje **AppKey**, což je klíč sloužící pro aktivaci OTAA zařízení. Finálním krokem je kliknutí na tlačítko **Register**.



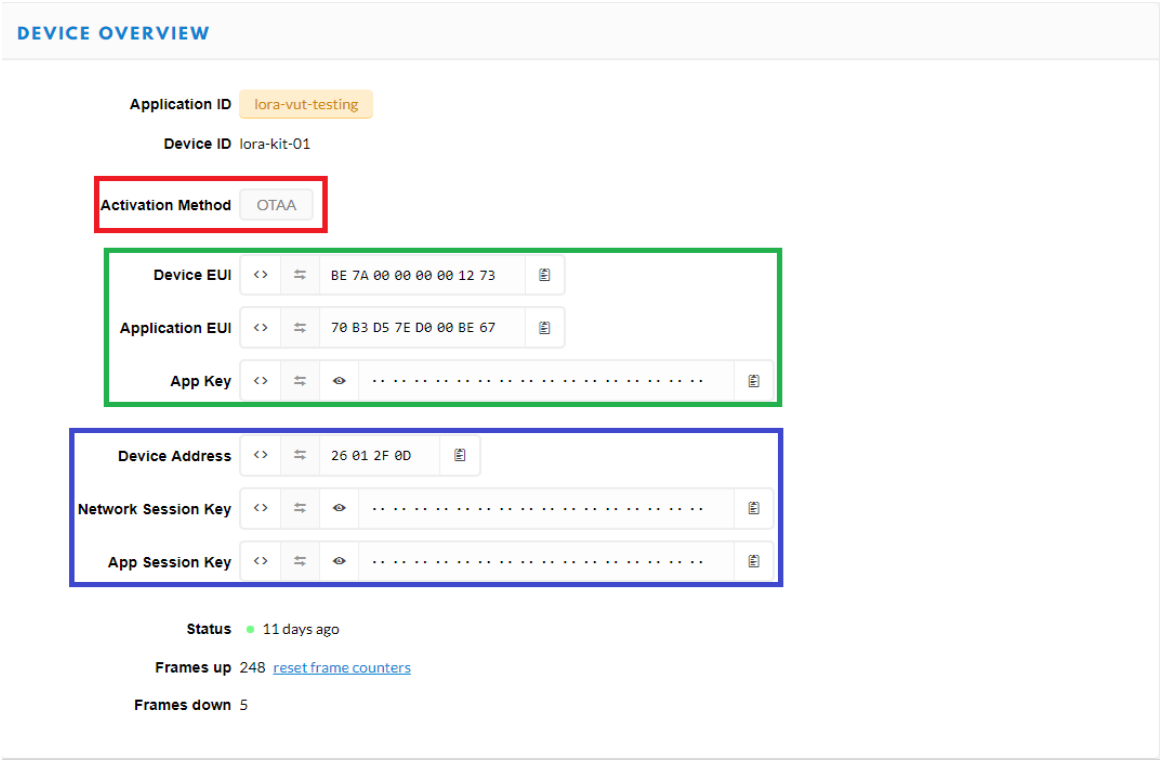
Obr. 8.18: Registrace zařízení



Obr. 8.19: Formulář registrace zařízení

Po vyplnění údajů jsou čidla v základu nastavená na aktivační metodu OTAA. Pokud chce uživatel přenastavit aktivační metodu na ABP, je tato možnost v sekci nastavení.

Obrázek 8.20 znázorňuje příklad nastaveného čidla. V tomto případě je použita aktivační metoda OTAA, která je ohraničena červeným rámečkem. V zeleném rámečku jsou parametry pro OTAA metodu, které je nutno nastavit pomocí programu v čidle. V modrém rámečku jsou pak parametry, které by bylo nutné nastavit v čidle za použití aktivační metody ABP.



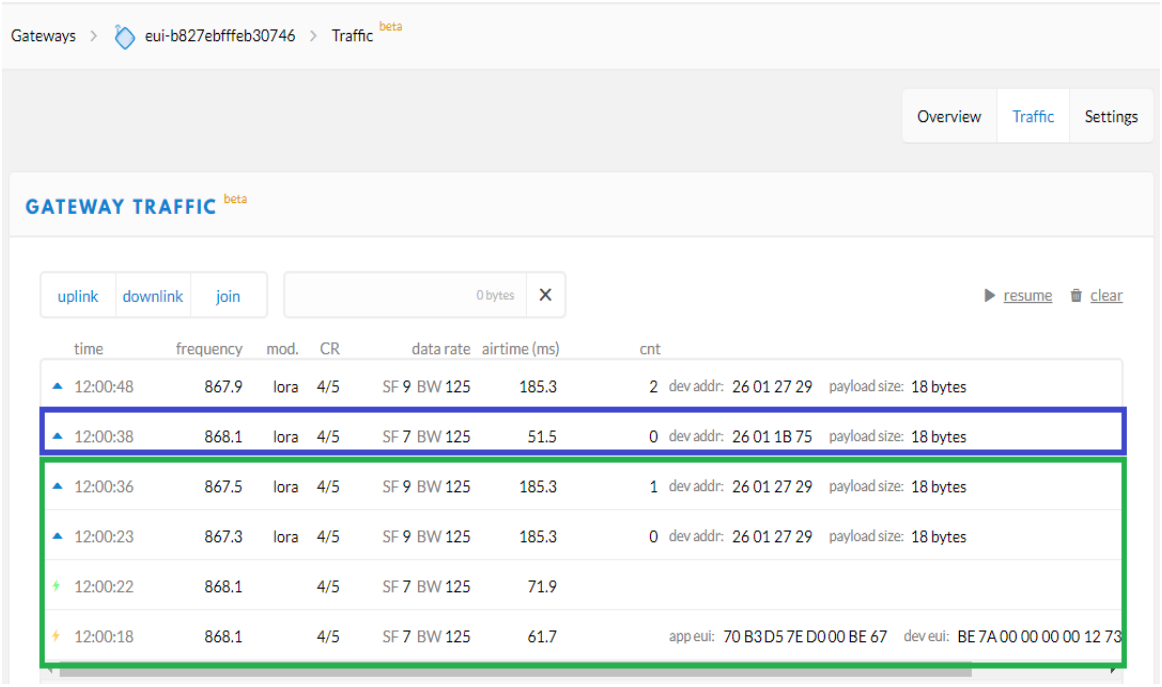
Obr. 8.20: Nastavení zařízení

Nyní je již vše přichystané k tomu, aby se čidla připojila k síti LoRaWAN. Posledním krokem je připojení čidel k odpovídajícímu napájecímu zdroji. Komunikaci na bráně je možné vidět na obrázku 8.21.

1. **Zprávy v zeleném rámečku**

   První zpráva zdola se symbolem žlutého blesku pochází od čidla s OTAA, v této zprávě je požadavek pro připojení k síti LoRaWAN, server na tuto zprávu patřičným způsobem reaguje a pokud veškeré nastavení souhlasí, je čidlo připojeno k síti viz zpráva se symbolem zeleného blesku. Tato zpráva pochází též od čidla a informuje správce systému, že aktivace zařízení proběhlo v pořádku. Třetí a čtvrtá zpráva zdola již obsahuje data daného čidla.

2. **Zprávy v modrém rámečku**
   Tato zpráva obsahuje již přímo data daného čidla, protože u ABP aktivace nedochází k žádným požadavkům o připojení.



Obr. 8.21: Komunikační provoz na bráně

9 REPLAY ÚTOK NA ABP ZAŘÍZENÍ

V této části je popsán hlavní cíl práce, a to zachycení paketů LoRa komunikace, jejich následné zfalšování a odeslání těchto zfalšovaných paketů opět na server. Popisu možných útoků na LoRaWAN sítě se věnovala kapitola 6. V průběhu této kapitoly jsou vysvětleny všechny kroky, které jsou k provedení tohoto útoku zapotřebí. Je však využito skutečnosti, že k bráně je umožněn fyzický přístup.

9.1 Paket forwarder

Jak bylo popsáno v kapitole 8, tak v instalaci je využit nativní paket forwarder portálu The Things Network **poly_pkt_fwd**. Paket forwarder má v programu brány velmi důležitou roli, protože je zodpovědný za předávání paketů z čidel, přes bránu až k serveru a naopak. Pro budoucí falšování paketů je tedy stěžejní zjistit, jaké asociace tento program vytváří. K tomu poslouží programy **ps** a **lsof**.

Program **ps** slouží k výpisu procesů a je možné ho obohatit o velké množství parametrů. Příkaz 9.1 použije parametry **axf**, které zajistí příslušné formátování a stromovou strukturu a **grep** vyfiltruje výpis, který se vztahuje pouze na **poly_pkt_fwd**. Z výpisu je patrné ID procesu 590, které bude využito později.

Výpis 9.1: Použití programu ps pro výpis procesů s filtrem

```
ttn@ttn-gateway:~ $ ps axf | grep poly_pkt_fwd
 2246 pts/0   S+     0:00            \_ grep --color=auto poly_pkt_fwd
  590 ?       Sl     8:39 \_ ./poly_pkt_fwd
```

Program **lsof** je potřeba nejdříve nainstalovat, protože se v systému v základu nenachází. K tomu poslouží příkaz 9.2.

Výpis 9.2: Instalace lsof

```
ttn@ttn-gateway:~ $ sudo apt-get install lsof
```

Následně je využito číslo procesu 590 pro zjištění asociací, které paket forwarder vytváří. Příkaz 9.3, kde parametr **-p** slouží k zadání čísla procesu, vypíše veškeré vytvořené asociace.

Výpis 9.3: Použití programu lsof

```
ttn@ttn-gateway:~ $ sudo lsof -p 590
COMMAND  PID USER  FD   TYPE    DEVICE SIZE/OFF  NODE NAME
poly_pkt_ 590 root cwd  DIR     179,2    4096 129326 /opt/ttn-
   gateway/bin
poly_pkt_ 590 root rtd  DIR     179,2    4096     2 /
poly_pkt_ 590 root txt  REG     179,2  117608 129318 /opt/ttn-
   gateway/packet_forwarder/poly_pkt_fwd/poly_pkt_fwd
poly_pkt_ 590 root mem  REG     179,2   75608  1969 /lib/arm-linux-
   gnueabihf/libresolv-2.24.so
poly_pkt_ 590 root mem  REG     179,2   18012  1943 /lib/arm-linux-
   gnueabihf/libnss_dns-2.24.so
poly_pkt_ 590 root mem  REG     179,2    9572  1948 /lib/arm-linux-
   gnueabihf/libnss_mdns4_minimal.so.2
poly_pkt_ 590 root mem  REG     179,2   38560  1944 /lib/arm-linux-
   gnueabihf/libnss_files-2.24.so
poly_pkt_ 590 root mem  REG     179,2 1234700  1905 /lib/arm-linux-
   gnueabihf/libc-2.24.so
poly_pkt_ 590 root mem  REG     179,2  127300  1966 /lib/arm-linux-
   gnueabihf/libpthread-2.24.so
poly_pkt_ 590 root mem  REG     179,2   26632  1970 /lib/arm-linux-
   gnueabihf/librt-2.24.so
poly_pkt_ 590 root mem  REG     179,2   21868 10410 /usr/lib/arm-
   linux-gnueabihf/libarmmem.so
poly_pkt_ 590 root mem  REG     179,2  138576  1858 /lib/arm-linux-
   gnueabihf/ld-2.24.so
poly_pkt_ 590 root  0r  CHR       1,3     0t0  1028 /dev/null
poly_pkt_ 590 root  1u  unix 0xb969a400    0t0 13523 type=STREAM
poly_pkt_ 590 root  2u  unix 0xb969a400    0t0 13523 type=STREAM
poly_pkt_ 590 root  3u  IPv4     11167     0t0   UDP
   192.168.255.100:60188->52.169.76.203:1700
poly_pkt_ 590 root  4u  IPv4     11169     0t0   UDP
   192.168.255.100:40323->52.169.76.203:1700
poly_pkt_ 590 root  5u  CHR     153,0     0t0  1676 /dev/spidev0.0
```

Z konce výpisu je patrné, že se z lokální IP adresy s náhodnými vysokými porty vytváří UDP spojení na veřejnou IP adresu serveru na port 1700, což je port pro LoRa komunikaci. Tyto informace jsou stěžejní pro další kroky.

9.2 Zachycení komunikace

Aby bylo možné později nějaký paket zfalšovat, je prvně nutné zachytit příslušnou LoRa komunikaci. K tomu může posloužit program **tcpdump**, který je nutné doinstalovat dle příkazu 9.4.

Výpis 9.4: Instalace tcpdump

```
ttn@ttn-gateway:~ $ sudo apt-get install tcpdump
```

Z výpisu 9.3 je patrné, že je zapotřebí zachytit komunikaci na portech 60188, 40323 nebo 1700. Zachytávání paketů na prvních dvou portech by bylo velmi nepraktické, protože tyto porty jsou náhodné a po každém restartu zařízení se vytvoří porty jiné. Komunikace bude tedy zachytávána na portu 1700, který je statický. Příkaz 9.5 zapříčiní zápis zachycených paketů do souboru 0001.pcap v hexadecimálním a ASCII formátu a zachytí všechny pakety s portem 1700, které projdou přes rozhraní eth0.

Výpis 9.5: Zachytávání paketů pomocí tcpdump

```
ttn@ttn-gateway:~ $ sudo tcpdump -w 0001.pcap -XX -i eth0 port 1700
```

Druhá možnost je napsání vlastního skriptu v jazyku Python, který bude zachytávání vykonávat. V systému Raspbian jsou v základu nainsalované obě verze, tudíž Python2 i Python3. Před samotným psaním skriptu je důležité nainstalovat instalátor balíčků pro Python3 – **pip3** pomocí příkazu 9.6.

Výpis 9.6: Instalace pip3

```
ttn@ttn-gateway:~ $ sudo apt-get install python3-pip
```

Nyní se nainstaluje program **scapy3k**. Scapy3k je program jazyku Python3, který umožňuje uživateli odesílat, zachytávat, vytvářet a analyzovat síťové pakety. To umožňuje konstrukci nástrojů, které dokáží sondovat, skenovat popřípadě napadnout síť. Jinými slovy, Scapy je výkonný interaktivní program pro manipulaci pakety. Je schopen vytvářet nebo dekódovat pakety mnoha protokolů, odesílat je, zachytit, odpovídat na požadavky a mnoho dalšího. Instalace je provedena pomocí příkazů 9.7 a 9.8.

Výpis 9.7: Instalace scapy3k

```
ttn@ttn-gateway:~ $ cd /tmp && git clone https://github.com/phaethon/
   scapy
```

Výpis 9.8: Instalace scapy3k

```
ttn@ttn-gateway:/tmp $ cd scapy && sudo python3 setup.py install
```

Nyní, po instalaci balíčku scapy, je již možné napsat vlastní skript v jazyce Python3. Obsah skriptu A.1 je k nalezení v přílohách A této práce. Ke skriptu jsou uvedeny odpovídající komentáře.

Spuštění skriptu se provede pomocí příkazu 9.9. Jakmile se zachytí 20 paketů, dle počítadla „count=20“ ve funkcni sniff na posledním řádku, tak se skript sám ukončí a vytvoří v adresáři **/home/ttn/sniff/** čtyři soubory ve formátu **pcap** viz výpis 9.10. Tyto soubory obsahují množství informací, které korespondují s vrstvami modelu TCP/IP.

Výpis 9.9: Spuštění skriptu sniffer.py

```
ttn@ttn-gateway:~/maliTools $ sudo python3 sniffer.py
```

Výpis 9.10: Spuštění skriptu sniffer.py

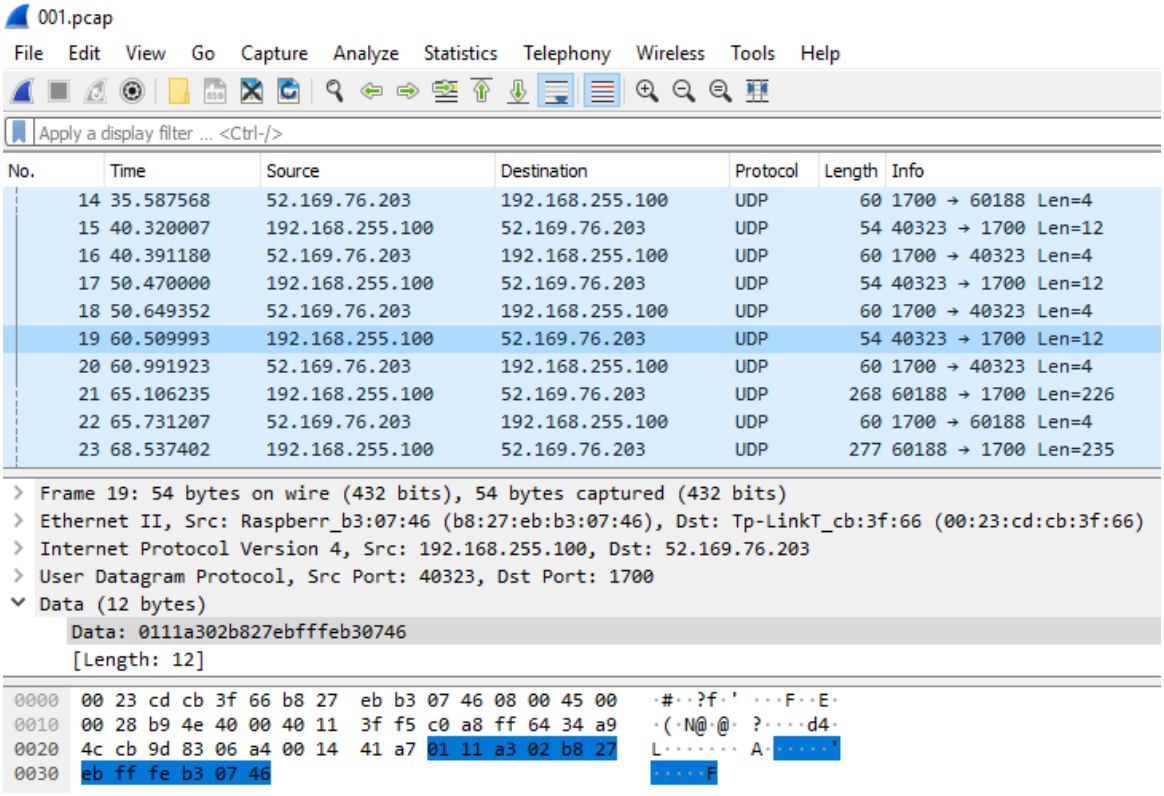
```
ttn@ttn-gateway:~/maliTools/sniff $ ls -l
total 24
-rw-r--r-- 1 root root 296 May 4 13:29 sniffedAppData.pcap
-rw-r--r-- 1 root root 380 May 4 13:29 sniffedEtherHeader.pcap
-rw-r--r-- 1 root root 352 May 4 13:29 sniffedIPHeader.pcap
-rw-r--r-- 1 root root 312 May 4 13:29 sniffedUDPHeader.pcap
-rw-r--r-- 1 ttn ttn 1666 May 4 13:28 sniffer.py
```

9.3 Rozbor zachycených paketů

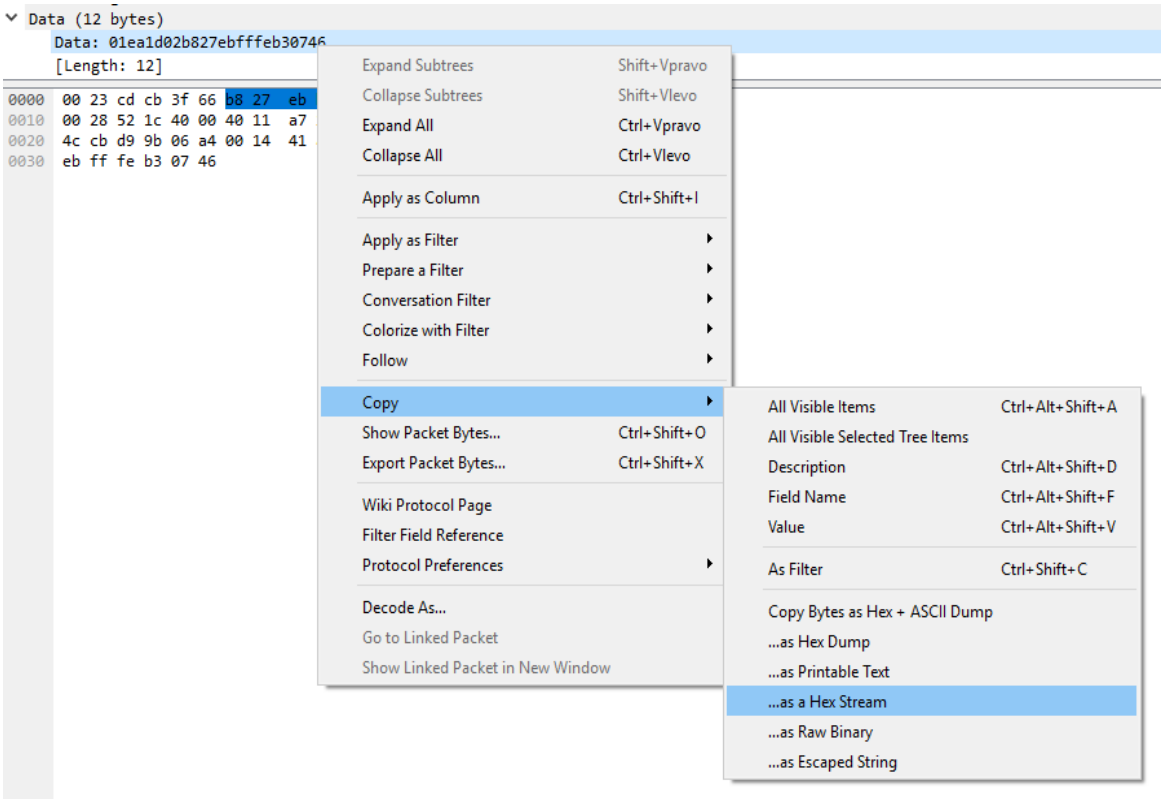
Aby bylo možné replikovat zachycené zprávy, je potřeba získat data ze zachycených paketů. Pomocí skriptu sniffer.py bylo zachyceno několik paketů LoRa komunikace v okamžiku, kdy zařízení aktivované metodou ABP už určitou dobu vysílalo a mělo tedy nenulovou hodnotu čítače rámců. Pro přehlednost byla ostatní zařízení odstavena. K přečtení a získání dat poslouží program **Wireshark**.

Obrázek 9.1 znázorňuje první zachycený paket (paket č. 19), který vyslala brána směrem k serveru. Znaky v modrém rámečku dole značí přenášena data. Pro provedení replay útoku jsou zapotřebí data, která odcházení z brány v pořadí velikosti 12 B, 226 B a 234-236 B. V první 12 B zprávě jsou první 4 B synchronizační data, která se s každou odezvou serveru inkrementují o 2 b a zbylých 8 B značí MAC adresu zařízení. Zpráva o délce 226 B (paket č. 21) obsahuje data z json formátu, která byla konfigurována v kapitole 8 a v poslední zprávě o délce 234-236 B (paket č. 23)

jsou už odesílána data ze senzoru. Všechna tato data, která jsou odesílána směrem k serveru jsou zkopírována a uložena na pozdější manipulaci. Zkopírování dat lze provést dle znázornění na obrázku 9.2.



Obr. 9.1: Znázornění zachycených paketů

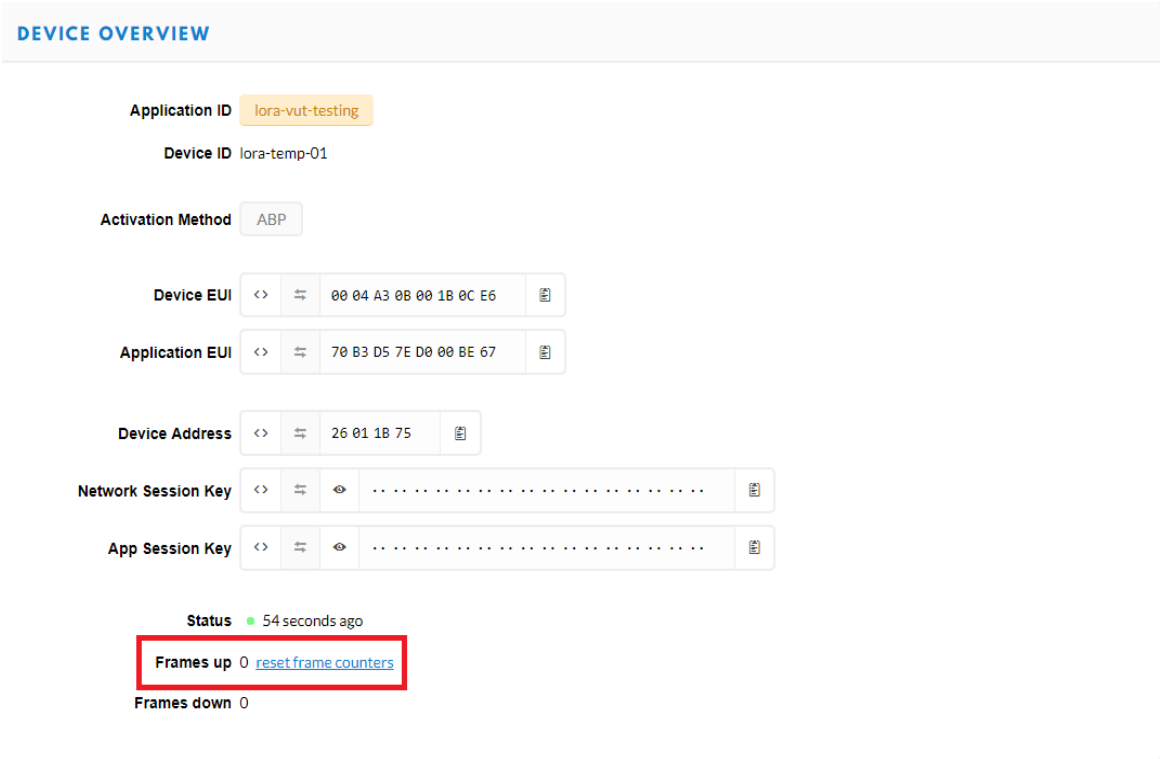


Obr. 9.2: Uložení dat z paketů

9.4 Provedení replay útoku

Dle informací specifikace LoRaWAN 1.0.2 je možné provést Replay útok na zařízení, které používá aktivační metodu ABP. Tato skutečnost je však možná pouze za předpokladu, že dojde k přetečení počítadla rámců, nebo když se zařízení resetuje například kvůli nedostatku napájecí energie. Poté další komunikace nemůže probíhat, protože server má zaznamenán jiný počet rámců, než zařízení, které začíná od 0. Replay útok je demonstrován pomocí skriptu **injection.py**, který odešle na server dříve odchycená data. Obsah skriptu A.2 je uveden v příloze A této práce.

Pro názornou ukázku je nyní zapotřebí ještě resetovat čítač rámců na serveru v menu příslušné aplikace a zařízení, jak ukazuje obrázek 9.3 a poté čidlo připojit k napájení.

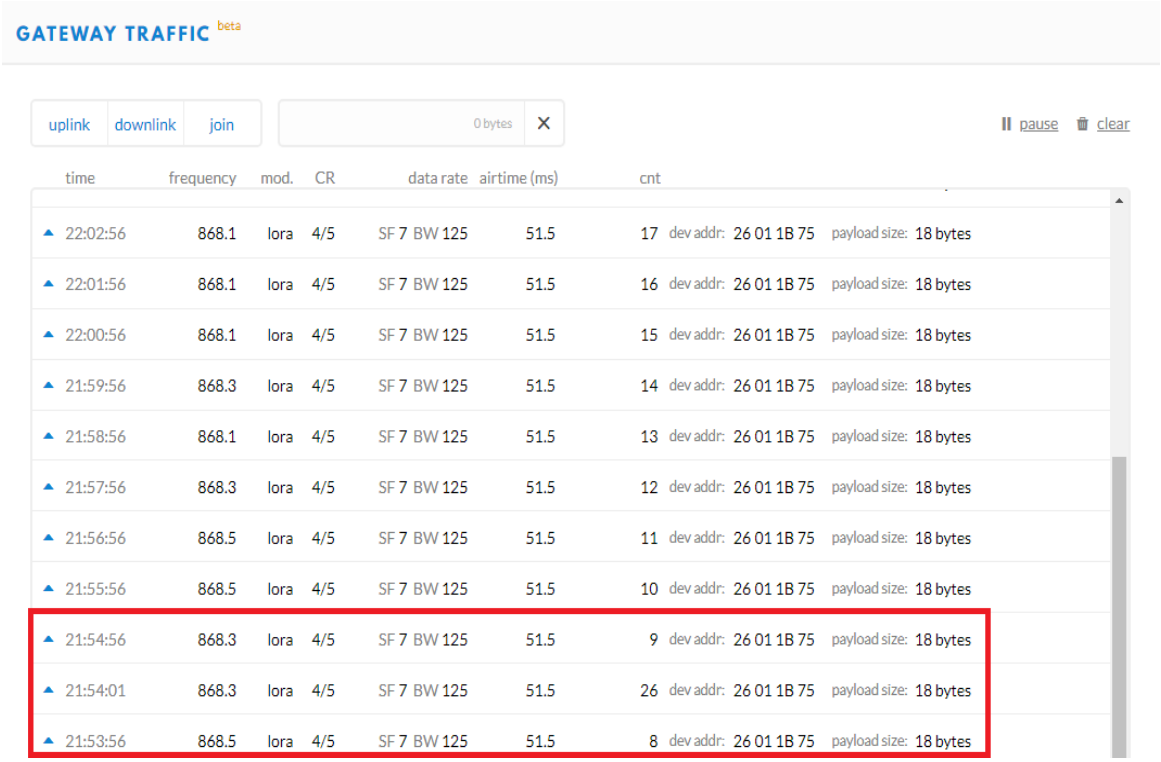


Obr. 9.3: Resetování čítače rámců

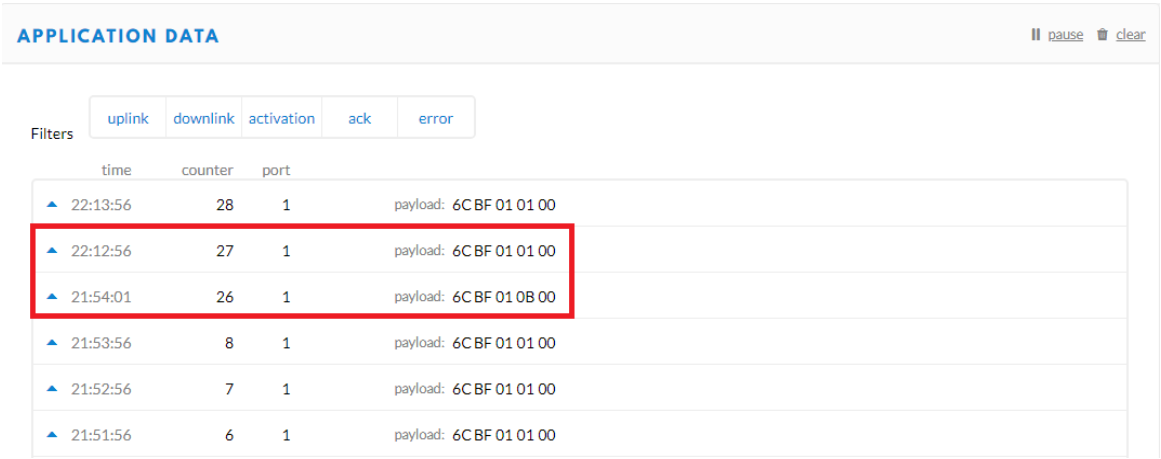
Po tomto kroku je spuštěn skript **injection.py** pomocí příkazu 9.11, který odešle dříve zachycená data znovu na server. Tato událost způsobí načtení rámce s vyšším koeficientem čítače. To znamená, že po započtení tohoto rámce je na bráně pozorovatelný i provoz rámců s nižším koeficientem čítače viz obrázek 9.4, ale aplikace veškerá mezilehlá data ignoruje do doby, než obdrží rámec s čítačem o 1 vyšší, než byl rámec falešný viz obrázek 9.5.

Výpis 9.11: Spuštění skriptu injection.py

```
ttn@ttn-gateway:~/maliTools $ python3 injection.py
WARNING: No route found for IPv6 destination :: (no default route?).
   This affects only IPv6
Data of packet has 12 B
Data of packet has 226 B
Data of packet has 235 B
```



Obr. 9.4: Provoz na bráně při replay útoku



Obr. 9.5: Pozastavená aplikační data při replay útoku

Skripty **sniffer.py** a **injection.py** dokázaly úspěšně zachytit a replikovat LoRa komunikaci a vytvořily z brány škodlivé zařízení, které je schopné ovlivnit bezpečný chod sítě. V příkladu, který byl v práci popsán je na čidlech použit interval jedné minuty, ale v praxi mají intervaly délku i jednoho dne. V takovém případě by

správce LoRaWAN sítě neměl z čidla žádná relevantní data po dobu několika dnů-týdnů. Práce tedy s patřičnými argumenty popisuje bezpečnostní rizika technologie LoRaWAN ve verzi specifikace 1.0.2, která je stále nasazená na velkém množství komerčních i nekomerčních síťových serverů.

10 ZÁVĚR

Diplomová práce se v teoretické části zabývala sítěmi s nízkým odběrem elektrické energie, které jsou zaměřené na Internet věcí. Mezi tyto sítě patří i LoRaWAN, která využívá pro komunikaci na dlouhý dosah modulaci LoRa, která je v práci také popsána.

Práce detailně popsala technologii LoRaWAN – topologii, architekturu, klíčové prvky, které jsou pro stavbu této sítě zapotřebí.

Dále se práce zabývala obecnou bezpečností Internetu věcí, popisem jednotlivých druhů útoků, možným bezpečnostním rizikům a zejména analýze bezpečnostních rizik sítě LoRaWAN.

Byly uvedeny známé proveditelné útoky na síť LoRaWAN a byl také popsán možný dopad těchto útoků.

V práci byl použit modul iC880A-SPI, který je rozšiřujícím modulem pro platformu Rapberry Pi a je v práci podrobně popsán.

V praktické části byly využity teoretické poznatky a bylo sestaveno fyzické zařízení LoRa brány. Toho bylo docíleno propojením modulu iC880A-SPI s platformou Raspberry Pi. V práci je uvedeno schematické zapojení i zobrazení fyzického provedení. Byla popsána kompletní konfigurace zařízení Raspberry Pi – instalace systému, vytvoření linuxového uživatele, přiřazení práv, nastavení síťování a vzdáleného přístupu, instalace programu LoRa brány, přiřazení brány k síťovému serveru. Dále byla popsána konfigurace ABP a OTAA senzorů, vytvoření aplikace, přiřazení čidel k aplikaci na aplikačním serveru, kompletní otestování všech komunikačních prvků.

Závěrečná část práce se zabývala realizací útoku probraného v teoretické části, konkrétně se jednalo o replay útok, který byl proveden za pomocí napsaných skriptů v jazyce Python. První skript **sniffer.py** slouží k zachycení komunikačních dat. Tato data bylo posléze možné použít v druhém skriptu **injection.py**. Spuštění skriptu provedlo replay útok, který je popsán v teoretické části práce.

Diplomová práce splnila zadání a v praktické části byla úspěšně potvrzena jedna ze slabých stránek současných IoT sítí a poukazuje tak na nezbytnost se aktivně na zabezpečení IoT podílet.

LITERATURA

[1] YANG, Wenjie, Mao WANG, Jingjing ZHANG, Jun ZOU, Min HUA, Tingting XIA a Xiaohu YOU. Narrowband Wireless Access for Low-Power Massive Internet of Things: A Bandwidth Perspective. IEEE Wireless Communications. 2017, 24(3), 138-145. DOI: 10.1109/MWC.2017.1600298. ISSN 1536-1284. Dostupné také z: http://ieeexplore.ieee.org/document/7934184/

[2] VEJLGAARD, Benny, Mads LAURIDSEN, Huan NGUYEN, Istvan Z. KOVACS, Preben MOGENSEN a Mads SORENSEN. Interference Impact on Coverage and Capacity for Low Power Wide Area IoT Networks. 2017 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2017, , 1-6. DOI: 10.1109/WCNC.2017.7925510. ISBN 978-1-5090-4183-1. Dostupné také z: http://ieeexplore.ieee.org/document/7925510/

[3] RATASUK, Rapeepat, Nitin MANGALVEDHE, Jorma KAIKKONEN a Michel ROBERT. Data Channel Design and Performance for LTE Narrowband IoT. 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall). IEEE, 2016, , 1-5. DOI: 10.1109/VTCFall.2016.7880951. ISBN 978-1-5090-1701-0. Dostupné také z: http://ieeexplore.ieee.org/document/7880951/

[4] NOLAN, Keith E., Wael GUIBENE a Mark Y. KELLY. An evaluation of low power wide area network technologies for the Internet of Things. 2016 International Wireless Communications and Mobile Computing Conference (IWCMC). IEEE, 2016, , 439-444. DOI: 10.1109/IWCMC.2016.7577098. ISBN 978-1-5090-0304-4. Dostupné také z: http://ieeexplore.ieee.org/document/7577098/

[5] BAI, Jialing a Guangliang REN. Polarized MIMO Slotted ALOHA Random Access Scheme in Satellite Network. IEEE Access. , 1-1. DOI: 10.1109/ACCESS.2017.2774247. ISSN 2169-3536. Dostupné také z: http://ieeexplore.ieee.org/document/8113455/

[6] A technical overview of LoRa® and LoRaWAN™. LoRa-alliance [online]. [cit. 2017-12-09]. Dostupné z: https://docs.wixstatic.com/ugd/eccc1a_ed71ea1cd969417493c74e4a13c55685.pdf

[7] DE CARVALHO SILVA, Jonathan, Joel J. P. C. RODRIGUES a Antonio M. ALBERTI. LoRaWAN — A low power WAN protocol for Internet of Things: A review and opportunities [online]. 1. IEEE: IEEE, 2017 [cit. 2017-12-09]. ISBN 978-953-290-071-2. Dostupné z: http://ieeexplore.ieee.org/document/8019271/

[8] LAVRIC, Alexandru a Valentin POPA. Internet of Things and LoRa™ Low-Power Wide-Area Networks: A survey. 2017 International Symposium on Signals, Circuits and Systems (ISSCS). IEEE, 2017, , 1-5. DOI: 10.1109/ISSCS.2017.8034915. ISBN 978-1-5386-0674-2. Dostupné také z: http://ieeexplore.ieee.org/document/8034915/

[9] RIZZI, Mattia, Paolo FERRARI, Alessandra FLAMMINI a Emiliano SISINNI. Evaluation of the IoT LoRaWAN Solution for Distributed Measurement Applications. IEEE Transactions on Instrumentation and Measurement. 2017, 66(12), 3340-3349. DOI: 10.1109/TIM.2017.2746378. ISSN 0018-9456. Dostupné také z: http://ieeexplore.ieee.org/document/8036410/

[10] AN1200.22: LoRa™ Modulation Basics [online]. ©2015 Semtech Corporation, May 2015 [cit. 2018-05-06]. Dostupné z: https://www.semtech.com/uploads/documents/an1200.22.pdf

[11] PATAVINA TECHNOLOGIES. LoRa PT System Architecture [online]. Patavina Technologies, 2017 [cit. 2017-12-09]. Dostupné z: http://www.patavinatech.com/pt/wp-content/uploads/LoRaPTSystemArchitecture.pdf

[12] LoRa-Alliance. LoRaWAN ™ 101: LoRaWAN ™ 101 [online]. [cit. 2018-05-16]. Dostupné z: https://docs.wixstatic.com/ugd/eccc1a_20fe760334f84a9788c5b11820281bd0.pdf

[13] ANDREA, Ioannis, Chrysostomos CHRYSOSTOMOU a George HADJICHRISTOFI. Internet of Things: Security vulnerabilities and challenges. 2015 IEEE Symposium on Computers and Communication (ISCC). IEEE, 2015, , 180-187. DOI: 10.1109/ISCC.2015.7405513. ISBN 978-1-4673-7194-0. Dostupné také z: http://ieeexplore.ieee.org/document/7405513/

[14] 10 Internet of Things Security Vulnerabilities [online]. [cit. 2017-12-12]. Dostupné z: http://blog.learningtree.com/10-internet-of-things-security-vulnerabilities/

[15] LoRaWAN: Vulnerability Analysis and Practical Expoitation [online]. Delft University of Technology, 2017 [cit. 2018-05-06]. Dostupné z: https://repository.tudelft.nl/islandora/object/uuid%3A87730790-6166-4424-9d82-8fe815733f1e. Master thesis. Delft University of Technology.

[16] WiMOD iC880A. Https://wireless-solutions.de [online]. [cit. 2017-12-09]. Dostupné z: https://wireless-solutions.de/images/stories/downloads/Radio%20Modules/iC880A/iC880A_Datasheet_V0_50.pdf

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

3GPP	The 3rd Generation Partnership Project
ABP	Activation By Personalization
ACK	Acknowledgement
ACL	Access Control List
ADR	Adaptive Data Rate
AES	Advanced Encryption Standard
ALOHA	ALOHA je protokol s náhodným přístupem, který je v poslední době využíván zejména v MTC komunikaci [5].
API	Application Programming Interface
BLE	Bluetooth Low Energy
BPSK	Binary Phase-Shift Keying
BT	Bandwith Time product
CIA	Confidentiality, Integrity, and Availability
CRC	Cyclic Redundancy Check
CSS	Chirp Spread Spectrum
DoS	Denial of Service
DDoS	Distributed Denial of Service
D-BPSK	Differential Binary Phase-Shift Keying
FIFO	First In First Out
FSK	Frequency-Shift Keying
FTDI	Future Technology Devices International
GFSK	Gaussian Frequency-Shift Keying
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HAL	Hardware Abstraction Layer
IoT	Internet of Things
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
JSON	JavaScript Object Notation
LED	Light-Emitting Diode
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wide Area Network
M2M	Machine-to-Machine
MAC	Medium Access Controll
MIC	Message Integrity Code

MTC	Machine Type Communication
NB-IoT	Narrow Band internet of Things
OFDMA	Orthogonal Frequency-Division Multiple Access
OTAA	Over-the-Air-Activation
OWASP	The Open Web Application Security Project
PC	Personal Computer
QPSK	Quadrature Phase-Shift Keying
RFID	Radio Frequency Identification
RSA	iniciály autorů Rivest, Shamir, Adleman
SD	Secure Digital
SF	Spreading Factor
SNR	Signal-to-noise ratio
SPI	Serial Peripheral Interface Bus
UART	Universal Asynchronous Receiver and Transmitter
UDP	User Datagram Protocol
UE	User Equipment
UNB	Ultra-Narrow Band
USB	Universal Serial Bus

SEZNAM PŘÍLOH

A PŘILOŽENÉ ZDROJOVÉ KÓDY

Výpis A.1: Obsah skriptu sniffer.py

```
#!/usr/bin/evn python3

#import konkretnich podbalicku z balicku scapy/scapy3k
try:
    from scapy.all import IP, UDP, sniff, Raw,
    PcapWriter, rdpcap, wrpcap
except:
    from scapy3k.all import IP, UDP, sniff, Raw,
    PcapWriter, rdpcap, wrpcap

#definice funkce print_summary s argumentem pkt
def print_summary(pkt):
#debugging pomoci pdb konzole
#    import pdb
#    pdb.set_trace()

    try:
        # definice promennych funkce print_summary(pkt)
        ip_src=pkt[IP].src  #zdrojova IP
        ip_dst=pkt[IP].dst  #cilova IP
        udp_sport=pkt[UDP].sport #zdrojovy port
        udp_dport=pkt[UDP].dport #cilovy port
        lport = 1700    #LoRa port
        IP_1 = '192.168.255.100' #privatni IP brany
        IP_2 = '52.169.76.203'  #verejna IP serveru

        # filtrovani IP a PORTu dle podminky
        if (((ip_src == IP_1) and (udp_dport == lport))
            or ((ip_dst == IP_2) and (udp_dport == lport))):
            print ("IP src " + str(ip_src) + " UDP sport "
            + str(udp_sport))

            print ("IP dst " + str(ip_dst) + " UDP dport "
            + str(udp_dport))

```

```
36              #zapis do souboru v danem adresari, obsah souboru
37              #odpovida nazvum
38              wrpcap('/home/ttn/sniff/sniffedEtherHeader.pcap',
39              pkt, append=True)
40              wrpcap('/home/ttn/sniff/sniffedIPHeader.pcap',
41              pkt.payload, append=True)
42              wrpcap('/home/ttn/sniff/sniffedUDPHeader.pcap',
43              pkt.payload.payload, append=True)
44              wrpcap('/home/ttn/sniff/sniffedAppData.pcap',
45              pkt.payload.payload.payload, append=True)
46      except:
47          pass
48  #do promenne packets se nacte vysledek volani funkce sniff
49  #s danymi parametry
50  packets=sniff(filter='ip',prn=print_summary, count = 20)
```

Výpis A.2: Obsah skriptu injection.py

```
#!/usr/bin/evn python3

#import balicku
from time import sleep
try:
    from scapy.all import IP, UDP, Raw, send, socket, StreamSocket
except:
    from scapy3k.all import IP, UDP, Raw, send, socket, StreamSocket

#definice funkce sendPacket s argumenty stream a data
def sendPacket(stream, data):
    #vytvoreni objektu dle tridy Raw z balicku scapy a prevod dat
    #z hexa na bytes
    ascapypacket=Raw(load=bytes.fromhex(data))
    #vypis delky odeslaneho paketu
    print ('Data of packet has',len(ascapypacket.getlayer(Raw)),'B')
    #volani metody send objektu stream, odeslani paketu
    stream.send(ascapypacket)
    #zpozdeni kazdeho paketu o 0.2 sekund
    sleep(0.2)
#hlavni cast programu
if __name__ == '__main__':
    #volani funkce socket ze tridy socket s argumenty AF_INET - IPv4,
    #SOCK_DGRAM - UDP datagram a ulozeni do promenne mysocket
    mysocket=socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    #volani metody connect objektu mysocket, parametrem jsou
    #destinacni IP a PORT
    mysocket.connect(('52.169.76.203', 1700))
    #vytvoreni objektu dle tridy StreamSocket
    mystream=StreamSocket(mysocket)

    packets = ['data 12B paketu',
               'data 226B paketu',
               'data 234-236B paketu']
    for p in packets:
        sendPacket(mystream, p)
```

B OBSAH PŘILOŽENÉHO CD

Na přiloženém CD se nachází zdrojové kódy, studie LoRa PT System Architecture, specifikace LoRaWAN, návod k použití senzoru společnosti SolidusTech a soubory se zachycenou LoRa komunikací.

```
Obsah CD
├── složka maliTools
│   ├── složka diagnostics
│   │   ├── soubor 0001.pcap
│   │   ├── soubor 0002.pcap
│   │   ├── soubor 0003.pcap
│   │   └── soubor 0004.pcap
│   ├── složka injection
│   │   └── Python skript injection.py
│   └── složka sniffing
│       ├── soubor sniffedAppData.pcap
│       ├── soubor sniffedEtherHeader.pcap
│       ├── soubor sniffedIPHeader.pcap
│       ├── soubor sniffedUDPHeader.pcap
│       └── Python skript sniffer.py
├── složka Materiály
│   ├── LoRaPTSystemArchitecture.pdf
│   ├── LoRaWAN1.0.1_d3.pdf
│   ├── LoRaWAN-Backend-Interfaces-v1.0.pdf
│   ├── LoRaWANRegionalParameters_v1.1rB-Final.pdf
│   ├── LoRaWANRegionalParametersv1.0.2_final_1944_1.pdf
│   ├── LoRaWAN-v1.1.pdf
│   └── NávodLoRaMiniUNI.pdf
└── složka SODAQ
    └── program teplota.ino
```